Free-hand Sketch Synthesis with Deformable Stroke Models

Int J Comput Vis
DOI 10.1007/s11263-016-0963-9

Free-Hand Sketch Synthesis with Deformable Stroke Models

Yi Li[1] · **Yi-Zhe Song**[1] · **Timothy M. Hospedales**[1,2] · **Shaogang Gong**[1]

Received: 9 October 2015 / Accepted: 30 September 2016

Abstract We present a generative model which can automatically summarize the stroke composition of free-hand sketches of a given category. When our model is fit to a collection of sketches with similar poses, it discovers and learns the structure and appearance of a set of coherent parts, with each part represented by a group of strokes. It represents both consistent (topology) as well as diverse aspects (structure and appearance variations) of each sketch category. Key to the success of our model are important insights learned from a comprehensive study performed on human stroke data. By fitting this model to images, we are able to synthesize visually similar and pleasant free-hand sketches.

Keywords Stroke analysis · Perceptual grouping · Deformable stroke model · Sketch synthesis

1 Introduction

Sketching comes naturally to humans. With the proliferation of touchscreens, we can now sketch effortlessly and ubiquitously by sweeping fingers on phones, tablets and smart watches. Studying free-hand sketches has thus become increasingly popular in recent years, with a wide spectrum of work addressing sketch recognition, sketch-based image retrieval, and sketching style and abstraction.

While computers are approaching human level on recognizing free-hand sketches (Eitz et al. 2012; Schneider and Tuytelaars 2014; Yu et al. 2015), their capability of synthesizing sketches, especially free-hand sketches, has not been fully explored. The main existing works on sketch synthesis are engineered specifically and exclusively for a single category: human faces. Albeit successful at synthesizing sketches, important assumptions are ubiquitously made that render them not directly applicable to a wider range of categories. It is often assumed that because faces exhibit quite stable structure (1) hand-crafted models specific to faces are sufficient to capture structural and appearance variations, (2) auxiliary datasets of part-aligned photo and sketch pairs are mandatory and must be collected and annotated (however labour intensive), (3) as a result of the strict data alignment, sketch synthesis is often performed in a relatively ad-hoc fashion, e.g., simple patch replacement. With a single exception that utilized professional strokes (rather than patches) (Berger et al. 2013), synthesized results resemble little the style and abstraction of free-hand sketches.

In this paper, going beyond just one object category, we present a generative data-driven model for free-hand sketch synthesis of diverse object categories. In contrast with prior art, (1) our model is capable of capturing structural and appearance variations without the handcrafted structural prior, (2) we do not require purpose-built datasets to learn from, but instead utilize publicly available datasets of free-hand sketches that exhibit no alignment nor part labeling and (3) our model fits free-hand strokes to an image via a detection process, thus capturing the specific structural and

Communicated by S.-C. Zhu.

✉ Yi Li
yi.li@qmul.ac.uk

Yi-Zhe Song
yizhe.song@qmul.ac.uk

Timothy M. Hospedales
t.hospedales@qmul.ac.uk; t.hospedales@ed.ac.uk

Shaogang Gong
s.gong@qmul.ac.uk

[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

[2] School of Informatics, The University of Edinburgh, Edinburgh, UK

Published online: 15 October 2016

appearance variation of the image and performing synthesis in free-hand sketch style.

By training on a few sketches of similar poses (e.g., standing horse facing left), our model automatically discovers semantic parts—including their number, appearance and topology—from stroke data, as well as modeling their variability in appearance and location. For a given sketch category, we construct a deformable stroke model (DSM), that models the category at a stroke-level meanwhile encodes different structural variations (deformable). Once a DSM is learned, we can perform image to free-hand sketch conversion by synthesizing a sketch with the best trade-off between an image edge map and a prior in the form of the learned sketch model. This unique capability is critically dependent on our DSM that represents enough stroke diversity to match any image edge map, while simultaneously modeling topological layout so as to ensure visual plausibility.

Building such a model automatically is challenging. Similar models designed for images either require intensive supervision (Felzenszwalb and Huttenlocher 2005) or produce imprecise and duplicated parts (Shotton et al. 2008; Opelt et al. 2006). Thanks to a comprehensive analysis into stroke data that is unique to free-hand sketches, we demonstrate how semantic parts of sketches can be accurately extracted with minimal supervision. More specifically, we propose a perceptual grouping algorithm that forms raw strokes into semantically meaningful parts, which for the first time synergistically accounts for cues specific to free-hand sketches such as stroke length and temporal drawing order. The perceptual grouper enforces part semantics within an individual sketch, yet to build a category-level sketch model, a mechanism is required to extract category-level parts. For that, we further propose an iterative framework that interchangeably performs: (1) perceptual grouping on individual sketches, (2) category-level DSM learning, and (3) DSM detection/stroke labeling on training sketches. Once learned, our model generally captures all semantic parts shared across one object category without duplication. An overview of our work is shown in Fig. 1, including both deformable stroke model learning and the free-hand sketch synthesis application.

The contribution of our work is threefold :

- A comprehensive and empirical analysis of sketch stroke data, highlighting the relationship between stroke length and stroke semantics, as well as the reliability of the stroke temporal order.
- A perceptual grouping algorithm based on stroke analysis is proposed, which for the first time synergistically accounts for multiple cues, notably stroke length and stroke temporal order.
- By employing our perceptual grouping method, a deformable stroke model is automatically learned in an iterative process. This model encodes both the common topology and the variations in structure and appearance of a given sketch category. Afterwards a novel and general sketch synthesis application is derived from the learned sketch model.

We evaluate our framework via user studies and experiments on two publicly available sketch datasets: (1) six diverse categories from non-expert sketches from the TU-Berlin dataset (Eitz et al. 2012) including: *horse, shark, duck, bicycle, teapot* and *face*, and (2) professional sketches of two *abstraction levels (90s and 30s; 's' is short for seconds indi-*

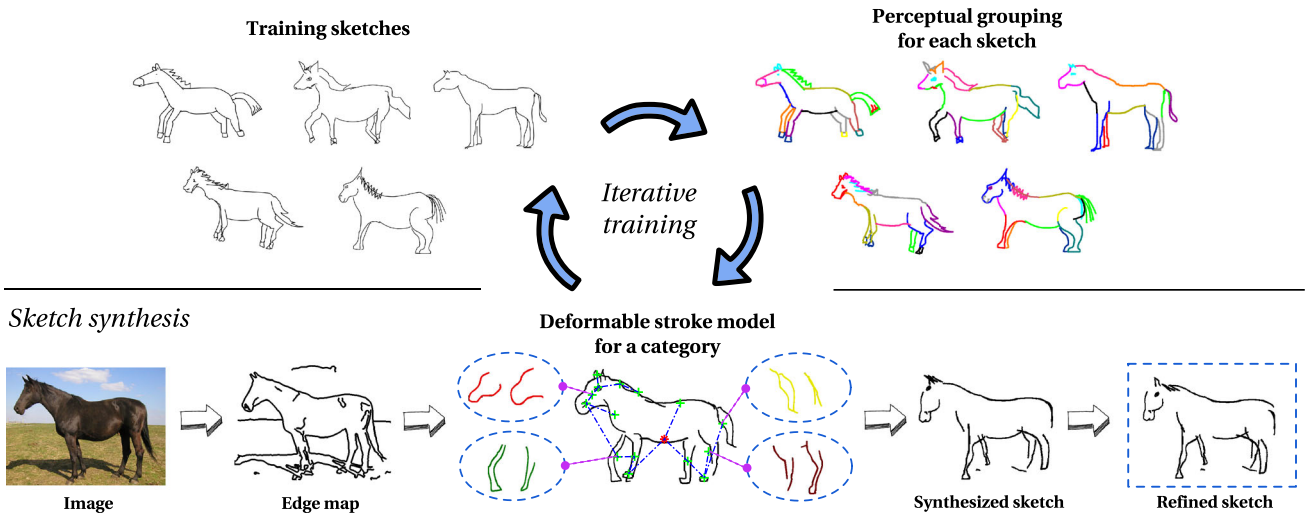

Fig. 1 An overview of our framework, encompassing deformable stroke model (DSM) learning and free-hand sketch synthesis for given images. To learn a DSM, (1) raw sketch strokes are grouped into semantic parts by perceptual grouping (semantic parts are not totally consistent across sketches); (2) a category-level DSM is learned on those semantic parts (category-level semantic parts are summarized and encoded); (3) the learned DSM is used to guide the perceptual grouping in the next iteration until convergence. When the DSM is obtained, we can synthesize sketches for a given image that are of a clear free-hand style, while being visually similar to the input image

cating the time used to compose the sketch) of two artists in the Disney portrait dataset (Berger et al. 2013).

2 Related Work

In this section, we start by reviewing several fields that generate sketch-like images and explaining why they are not suitable for general purpose free-hand sketch synthesis. We also offer reviews on the modelling methods that either inspired our deformable stroke model or share close resemblance. Towards the end, we review recent progress on sketch stroke analysis and sketch segmentation, both of which are important parts of the proposed free-hand sketch synthesis framework.

2.1 Photo to Sketch Stylization

Plenty of works from the non-photorealistic animation and rendering (NPAR) community can produce sketch-like results for 2D images or 3D models. Several works (Gooch et al. 2004; Kang et al. 2007; Kyprianidis and Döllner 2008; Winnemöller 2011) acknowledged that the Difference-of-Gaussians (DoG) operator could produce aesthetically more pleasing edges than traditional edge detectors, e.g. Canny (Canny 1986), and employed it to synthesize line drawings and cartoons. We offer comparisons with two representative DoG-oriented techniques in this paper: the flow-based DoG (FDoG) (Kang et al. 2007) that uses edge tangent flow (ETF) to offer edge direction guidance for DoG filtering (originally computed isotropically) and the variable thresholding DoG (XDoG) (Winnemöller 2011) that introduces several additional parameters to the filtering function in order to augment the remit of rendering styles. Quite a large body of the literature (Cole et al. 2008; DeCarlo et al. 2003; Judd et al. 2007; Grabli et al. 2010) studied the problem of generating line drawings from 3D models. Yet in contrast to synthesizing from 2D images, 3D models have well-defined structures and boundaries, which make the generation process much easier and less sensitive to noise. (Liu et al. 2014) attempted to simulate human sketching of 3D objects. They decomposed the sketching process into several fundamental phases, and a multi-phase framework was proposed to animate the sketching process and generate some realistic and visually plausible sketches. Generally speaking, although NPAR works share a high aesthetic standard, the generated images are still more realistic than free-hand sketch style. Severe artifacts are also hard to avoid at the presence of complicated textures.

Some perceptual organization and contour detection works also can generate sketch-like images that are abstract representations of the original images. (Guo et al. 2007) proposed a mid-level image representation named primal sketch. To generate such a primal sketch representation, a dictionary of image primitives was learned and Markov random fields were used to enforce the Gestalt (Koffka 1935) organization of image primitives. Qi et al. (2013) proposed a similar approach to extract a sketch from an image. Rather than learn a dictionary of primitives, they directly used long straight contours as primitives and employed a Gestalt grouper to form contour groups among which some prominent ones were kept to compose the final result. Ren et al. (2008) looked into the statistics of human-marked boundaries and observed power law distributions that were often associated with scale invariance. Based on the observation, a scale-invariant representation composed of piecewise linear segments was proposed and some probabilistic models were built to model the curvilinear continuity. Arbelaez et al. (2011) investigated both contour detection and image segmentation. Their gPb contour detector employed local cues computed with gradient operators and global information obtained by spectral clustering. They also reduced image segmentation to contour detection by proposing a method to transform any contour detection result into a hierarchical region tree. By replacing hand-crafted gradient features with Sparse Code Gradients (SCG) that were using patch representations automatially learned through sparse coding, Ren and Bo (2012) achieved state-of-the art contour detection performance. Recently, Lim et al. (2013) learned mid-level image features called sketch tokens by clustering patches from hand drawn contours in images. A random forest classifier (Breiman 2001) was then trained to assign the correct sketch token to a novel image patch. They achieved quite competitive contour detection performance at very low computational cost. We also include it in our comparison experiment. These works could achieve decent abstraction on images, but are still weak at dealing with artifacts and noise.

Data-driven approaches have been introduced to generate more human-like sketches, exclusively for one object category: human faces. Chen et al. (2002) and Liang et al. (2002) took simple exemplar-based approachs to synthesize faces and used holistic training sketches. Wang and Tang (2009) and Wang et al. (2012) decomposed training image-sketch pairs into patches, and trained a patch-level mapping model. All the above face synthesis systems work with professional sketches and assume perfect alignment across all training and testing data. As a result, patch-level replacement strategies are often sufficient to synthesize sketches. Moving onto free-hand sketches, Berger et al. (2013) directly used strokes of a portrait sketch dataset collected from professional artists, and learned a set of parameters that reflected style and abstraction of different artists. They achieved this by building artist-specific stroke libraries and performing a stroke-level study accounting for multiple characteristics. Upon synthesis, they first converted image edges into vector curves according to a chosen style, then replaced them with human strokes measuring shape, curvature and length.

Although these stroke-level operations provided more freedom during synthesis, the assumption of rigorous alignment is still made (manually fitting a face-specific mesh model to images and sketches), making extension to wider categories non-trivial. Their work laid a solid foundation for future study on free-hand sketch synthesis, yet extending it to many categories presents three major challenges: (1) sketches with fully annotated parts or feature points are difficult and costly to acquire, especially for more than one category; (2) intra-category appearance and structure variations are larger in categories other than faces, and (3) a better means of model fitting is required to account for noisier edges. In this paper, we design a model that is flexible enough to account for all these highlighted problems.

2.2 Part or Contour/Stroke Modeling Methods

In the early 1990s, Saund (1992) had already studied to learn a shape/sketch representation that could encode geometrical structure knowledge of a specific shape domain. A shape vocabulary called constellations of shape tokens was learned and maintained in a Scale-Space Blackboard. Similar configurations of shape tokens that were deformation variations were jointly described by a scheme named dimensionality-reduction.

The And-Or graph is a hierarchical-compositional model which has been widely applied for sketch modeling. An And-node indicates a decomposition of a configuration or sub-configuration by its children, while an Or-node serves as a switch among alternative sub-configurations. Both the part appearance and structure variations can be encoded in the And-Or graph. Chen et al. (2006) employed this model to compose clothes sketches, based on manually separated sketch clothes parts. Xu et al. (2008) employed this model to reconstruct face photos at multiple resolutions and generate cartoon facial sketches with different levels of detail. They particularly arranged the And-Or graph into three layers with each layer having the independent ability to generate faces at a specific resolution, and therefore addressed multiple face resolutions. While the above two works are both tailored for a specific category, Wu et al. (2010) proposed an active basis model, which can also be seen as an And-Or graph, and can be applied to general categories. The active basis model consists of a set of Gabor wavelet elements which look like short strokes and can slightly perturb their locations and orientations to form different object variations. A shared sketch algorithm and a computational architecture of sum-max maps were employed for model learning and model recognition respectively. Our model in essence is also an And-Or graph with an And-node consisting the parts and Or-nodes encoding stroke exemplars. Our model learning and detection share resemblance to the above works but dramatically differ in that we learn our model from processed real human strokes and do not ask for any part-level supervision. In our experiments, we also compare with the active basis model (Wu et al. 2010).

Our model is mostly inspired by contour (Shotton et al. 2008; Opelt et al. 2006; Ferrari et al. 2010; Dai et al. 2013) and pictorial structure (Felzenszwalb and Huttenlocher 2005) models. Both have been shown to work well in the image domain, especially in terms of addressing holistic structural variation and noise robustness. The idea behind contour models is learning object parts directly on edge fragments. And a by-product of the contour model is that via detection an instance of the model will be left on the input image. Despite being able to generate sketch-like instances of the model, the main focus of that work is on object detection, therefore synthesized results do not exhibit sufficient aesthetic quality. Major drawbacks of contour models in the context of sketch synthesis are: (1) duplicated parts and missing details as a result of unsupervised learning, (2) rigid star-graph structure and relatively weak detector are not good at modeling sophisticated topology and enforcing plausible sketch geometry, and (3) inability to address appearance variations associated with local contour fragments. On the other hand, pictorial structure models are very efficient at explicitly and accurately modeling all mandatory parts and their spatial relationships. They work by using a minimum spanning tree and casting model learning and detection into a statistical maximum a posteriori (MAP) framework. However the favorable model accuracy is achieved at the cost of supervised learning that involves intensive manual labelling. The deformable part-based model (DPM) (Felzenszwalb et al. 2010), was proposed later on to improve pictorial structures' practical value on some very challenging datasets, e.g., PASCAL VOC (Everingham et al. 2007). Mixture models were included to address significant variations in one category, and a discriminative latent SVM was proposed for training models using only object bounding boxes as supervision. Although more powerful, the DPM framework involved too many engineering techniques for more efficient model learning and inference. Therefore, we choose to stick to the original pictorial structure approach while focusing on the fundamental concepts necessary for modeling sketch stroke data. By integrating pictorial structure and contour models, we propose a deformable stroke model that: (1) employs perceptual grouping and an iterative learning scheme, yielding accurate models with minimum human effort, (2) customizes pictorial structure learning and detection to address the more sophisticated topology possessed by sketches and achieve more effective stroke to edge map registration, and (3) augments contour model parts from just one uniform contour fragment to multiple stroke exemplars in order to capture local appearance variations.

2.3 Stroke Analysis

Despite the recent surge in sketch research, stroke-level analysis of human sketches remains sparse. Existing studies (Eitz et al. 2012; Berger et al. 2013; Schneider and Tuytelaars 2014) have mentioned stroke ordering, categorizing strokes into types, and the importance of individual strokes for recognition. However, a detailed analysis has been lacking especially towards: (1) level of semantics encoded by human strokes, and (2) the temporal sequencing of strokes within a given category.

Eitz et al. (2012) proposed a dataset of 20,000 human sketches and offered anecdotal evidence towards the role of stroke ordering. Fu et al. (2011) claimed that humans generally sketch in a hierarchical fashion, i.e., contours first, details second. Yet as can be seen later in Sect. 2.3, we found this does not always hold, especially for non-expert sketches. More recently, Schneider and Tuytelaars (2014) touched on stroke importance and demonstrated empirically that certain strokes are more important for sketch recognition. While interesting, none of the work above provided means of modeling stroke ordering/saliency in a computational framework, thus making potential applications unclear. Huang et al. (2014) was first in actually using temporal ordering of strokes as a soft grouping constraint. Similar to them, we also employ stroke ordering as a cost term in our grouping framework. Yet while they only took the temporal order grouping cue as a hypothesis, we move on to provide solid evidence to support its usage.

A more comprehensive analysis of strokes was performed by Berger et al. (2013) aiming to decode the style and abstraction of different artists. They claimed that stroke length correlates positively with abstraction level, and in turn categorized strokes into several types based on their geometrical characteristics. Although insightful, their analysis was constrained to a dataset of professional portrait sketches, whereas we perform an in-depth study into non-expert sketches of many categories as well as the professional portrait dataset and we specifically aim to understand stroke semantics rather than style and abstraction.

2.4 Part-Level Sketch Segmentation

Few works so far considered part-level sketch segmentation. Huang et al. (2014) worked with sketches of 3D objects, assuming that sketches do not possess noise or over-sketching (obvious overlapping strokes). Instead, we work on free-hand sketches where noise and over-sketching are pervasive. Qi et al. (2015) cast the edge segmentation problem into a graph cuts framework, and utilized a ranking strategy with two Gestalt principles to construct the edge graph. However, their method cannot control the size of stroke groups which is essential for obtaining meaningful sketch parts. Informed by a stroke-level analysis, our grouper not only uniquely considers temporal order and several Gestalt principles, but also controls group size to ensure semantic meaningfulness. Beside applying it on individual sketches, we also integrate the grouper with stroke model learning to achieve across-category consistency.

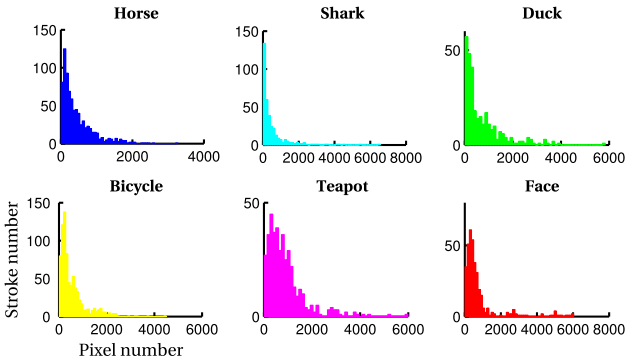

Fig. 2 Histograms of stroke lengths of six non-expert sketch categories. (*x-axis* the size of stroke in pixels; *y-axis* number of strokes in the category)

3 Stroke Analysis

In this section we perform a full analysis on how stroke-level information can be best used to locate semantic parts of sketches. In particular, we look into (1) the correlation between stroke length and its semantics as an object part, i.e., what kind of strokes do object parts correspond to, and (2) the reliability of temporal ordering of strokes as a grouping cue, i.e., to what degree can we rely on temporal information of strokes. We conduct our study on both non-expert and professional sketches: (1) six diverse categories from non-expert sketches from the TU-Berlin dataset (Eitz et al. 2012) including: *horse, shark, duck, bicycle, teapot* and *face*, and (2) professional sketches of two *abstraction levels (90s and 30s)* of *artist A* and *artist E* in the Disney portrait dataset (Berger et al. 2013).

3.1 Semantics of Strokes

On the TU-Berlin dataset, we first measure stroke length statistics (quantified by pixel count) of all six chosen categories. Histograms of each category are provided in Fig. 2. It can be observed that despite minor cross-category variations, distributions are always long-tailed: most strokes being shorter than 1000 pixels, with a small proportion exceeding 2000 pixels. We further divide strokes into 3 groups based on length, illustrated by examples of 2 categories in Fig. 3a. We can see that (1) medium-sized strokes tend to exhibit semantic parts of objects, (2) the majority of short strokes (e.g., <1000 px;

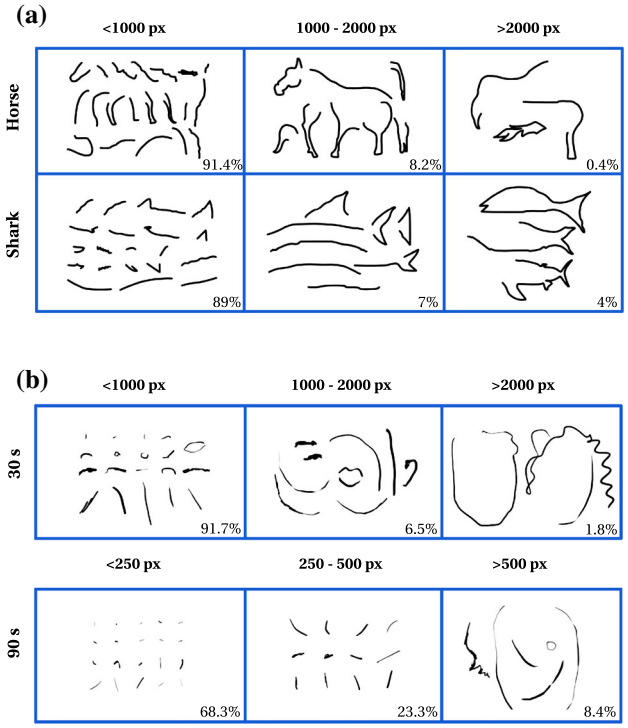

Fig. 3 Example strokes of each size group. **a** 2 categories in TU-Berlin dataset. **b** 2 levels of abstraction from artist A in Disney portrait dataset. The proportion of each size group in the given category is indicated in the bottom-right corner of each cell

'px' is short for pixels) are too small to correspond to a clear part, and (3) long strokes (e.g., >2000 px) lose clear meaning by encompassing more than one semantic part.

These observations indicate that, ideally, a stroke model can be directly learned on strokes from the medium length range. However, in practice, we further observe that people tend to draw very few medium-sized strokes (length correlates negatively with quantity as seen in Fig. 2), making them statistically insignificant for model learning. This is apparent when we look at percentages of strokes in each range, shown towards bottom right of each cell in Fig. 2. We are therefore motivated to propose a perceptual grouping mechanism that counters this problem by grouping short strokes into longer chains that constitute object parts (e.g., towards the medium range in the TU-Berlin sketch dataset). We call the grouped strokes representing semantic parts as semantic strokes. Meanwhile, a cutting mechanism is also employed to process the few very long strokes into segments of short and/or medium length, which can be processed by perceptual grouping afterwards.

On the Disney portrait dataset, a statistical analysis of strokes similar to Fig. 2 was already conducted by the original authors and the stroke length distributions are quite similar to ours. From example strokes in each range in Fig. 3b, we can see for sketches of the 30s level the situation is similar to the TU-Berlin dataset where most semantic strokes are clustered within the middle length range (i.e., 1000–2000 px) and the largest group is still the short strokes. As already claimed in (Berger et al. 2013) and also reflected in the bottom row of Fig. 3b, stroke lengths across the board reduce significantly as abstraction level goes down to 90s. This suggests that, for the purpose of extracting semantic parts, a grouping framework is even more necessary for professional sketches where individual strokes convey less semantic meaning.

3.2 Stroke Ordering

Another previously under-studied cue for sketch understanding is the temporal ordering of strokes, with only a few studies exploring this (Fu et al. 2011; Huang et al. 2014). Yet these authors only hypothesized the benefits of temporal ordering without critical analysis a priori. In order to examine if there is a consistent trend in holistic stroke ordering (e.g., if long strokes are drawn first followed by short strokes), we color-code length of each stroke in Fig. 4 where: each sketch is represented by a row of colored cells, ordering along the x-axis reflects drawing order, and sketches (rows) are sorted in ascending order of number of constituent strokes. For ease of interpretation, only 2 colors are used for the color-coding. Strokes with above average length are encoded as yellow and those with below average as cyan.

From Fig. 4 (1st and 2nd rows), we can see that non-expert sketches with fewer strokes tend to contain a bigger proportion of longer strokes (greater yellow proportion in the upper rows), which matches the claim made by (Berger et al. 2013). However, there is not a clear trend in the ordering of long and short strokes across all the categories. Although clearer trend of short strokes following long strokes can be observed in few categories, e.g., *shark* and *face*, and this is due to these categories' contour can be depicted by very few long and simple strokes. In most cases, long and short strokes appear interchangeably at random. Only in the more abstract sketches (upper rows), we can see a slight trend of long strokes being used more towards the beginning (more yellow on the left). This indicates that average humans draw sketches with a random order of strokes of various lengths, instead of a coherent global order in the form of a hierarchy (such as long strokes first, short ones second). In Fig. 4 (3rd row), we can see that artistic sketches exhibit a clearer pattern of a long stroke followed by several short strokes (the barcode pattern in the figure). However, there is still not a dominant trend that long strokes in general are finished before short strokes. This is different from the claim made by Fu et al. (2011), that most drawers, both amateurs and professionals, depict objects hierarchically. In fact, it can also be observed from Fig. 5 that average people often sketch objects part by part other than hierarchically. However the ordering of how parts are drawn appears to be random.

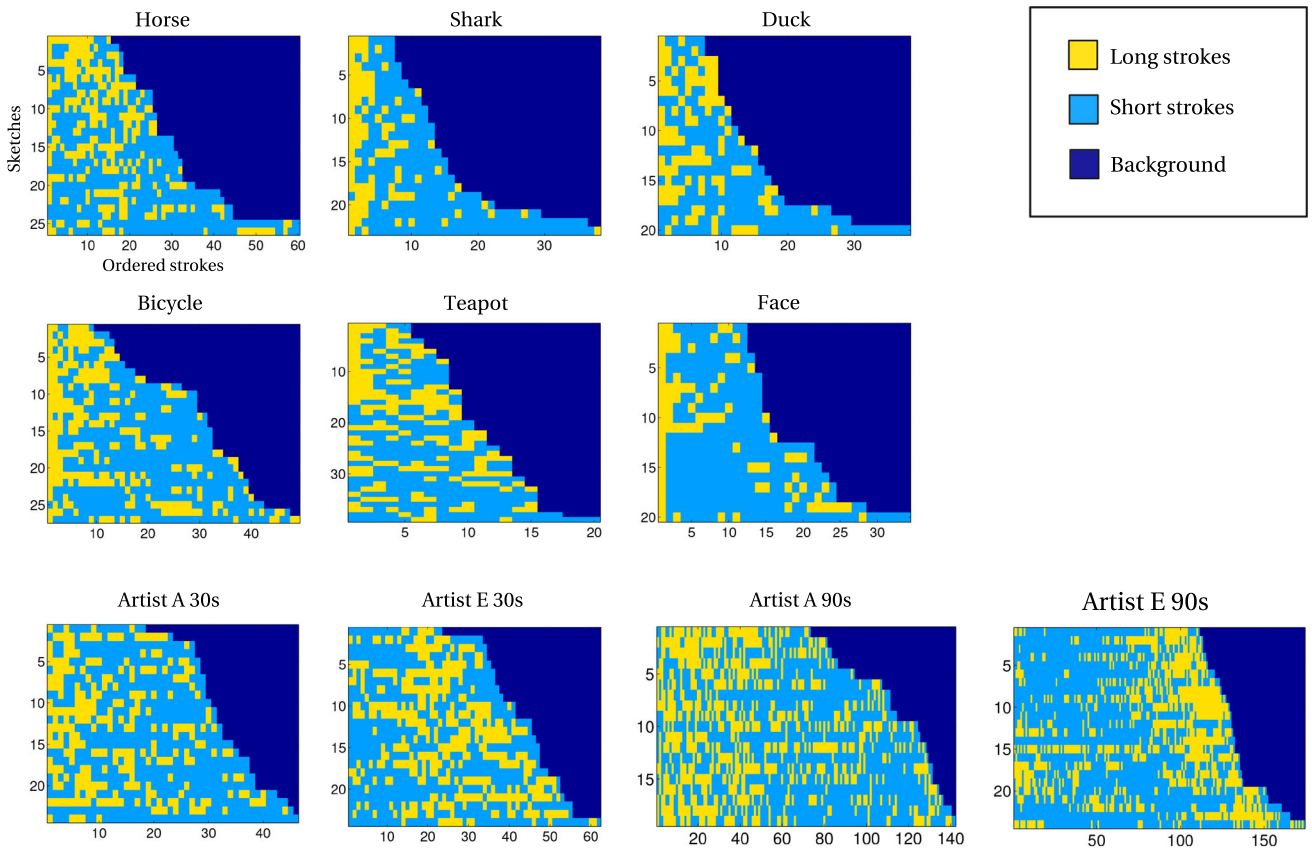

Fig. 4 Exploration of stroke temporal order. Subplots represent 10 categories: *horse*, *shark*, *duck*, *bicycle*, *teapot* and *face* of TU-Berlin dataset and 30s and 90s levels of *artist A* and *artist E* in Disney portrait dataset. *x-axis* shows stroke order and *y-axis* sketch samples, so each cell of the matrices is a stroke. Sketch samples are sorted by their number of strokes (abstraction). Shorter than average strokes are *yellow*, longer than average strokes are cyan

Although stroke ordering shows no global trend, we found that local stroke ordering (i.e., strokes depicted within a short timeframe) does possess a level of consistency that could be useful for semantic stroke grouping. Specifically, we observe that people tend to draw a series of consecutive strokes to depict one semantic part, as seen in Fig. 5. The same hypothesis was also made by Huang et al. (2014), but without clear stroke-level analysis beforehand. Later, we will demonstrate via our grouper how local temporal ordering of strokes can be modeled and help to form semantic strokes.

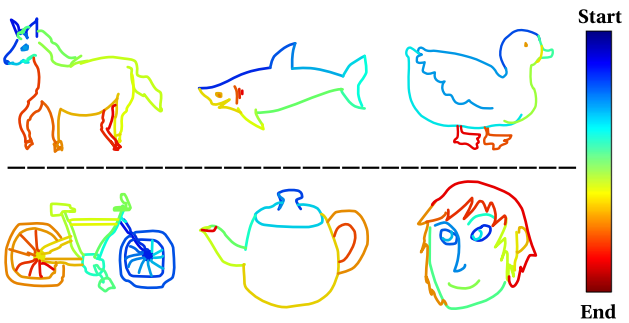

Fig. 5 Stroke drawing order encoded by color (starts from blue and ends at *red*). Object parts tend to be drawn with sequential strokes

4 A Deformable Stroke Model

From a collection of sketches of similar poses within one category, we can learn a generative deformable stroke model (DSM). In this section, we first formally define DSM and the Bayesian framework for model learning and model detection. Then, we offer detailed demonstration of the model learning process, the model detection process and the iterative learning scheme.

4.1 Model Definition

Our DSM is an undirected graph of n semantic part clusters: $G = (V, E)$. The vertices $V = \{v_1, \ldots, v_n\}$ represent category-level semantic part clusters, and pairs of semantic part clusters are connected by an edge $(v_i, v_j) \in E$ if their locations are closely related. The model is parameterized by

$\theta = (u, E, c)$, where $u = \{u_1, \ldots, u_n\}$, with $u_i = \{s_i^a\}_{a=1}^{m_i}$ representing m_i semantic stroke exemplars of the semantic part cluster v_i; E encodes pairwise part connectivity; and $c = \{c_{ij} | (v_i, v_j) \in E\}$ encodes the relative spatial relations between connected part clusters. We do not model the absolute location of each cluster for the purpose of generality. For efficient inference, we require the graph to form a tree structure and specifically we employ the minimum spanning tree (MST) in this paper. An example *shark* DSM illustration with full part clusters is shown in Fig. 11 (and a partial example for *horse* is already shown in Fig. 1), where the green crosses are the vertices V and the blue dashed lines are the edges E. The part exemplars u_i are highlighted in blue dashed ovals.

To learn such a DSM and employ it for sketch synthesis through object detection, we need to address 3 problems: (1) learning a DSM from examples, (2) sampling multiple good matches from an image, and (3) finding the best match of the model to an image. All these problems can be solved within the statistical framework described below. Let $F = \{(s_i, l_i)\}_{i=1}^n$ be a configuration of the DSM, indicating that exactly one stroke exemplar s_i is selected in each cluster and placed at location l_i. And Let I indicate the image. Then, the distribution $p(I|F, \theta)$ models the likelihood of observing an image given a learned model and a particular configuration. The distribution $p(F|\theta)$ models the prior probability that a sketch is composed of some specified semantic strokes with each stroke at a particular location. In the end, the posterior distribution $p(F|I, \theta)$ models the probability of a configuration given the image I and the DSM parameterized by θ. The posterior then can be written with Bayes' rule into:

$$p(F|I, \theta) \propto p(I|F, \theta) p(F|\theta) \tag{1}$$

Under this statistical framework, (1) the model parameter θ can be learned from training data using maximum likelihood estimation (MLE); (2) the posterior provides a path to sample multiple model candidates rather than just the best match; (3) finding the best match can be formed into a maximum a posteriori (MAP) estimation problem which can finally be cast as an energy minimization problem, as discussed in Sect. 4.3.2.

For the likelihood of seeing an image given a specified configuration, similarly to Felzenszwalb and Huttenlocher (2005), we approximate it with the product of the likelihoods of the semantic stroke exemplars/clusters,

$$p(I|F, \theta) = p(I|F) \propto \prod_{i=1}^{n} p(I|s_i, l_i). \tag{2}$$

θ is omitted since F has already encoded the selected stroke exemplars s_i. This approximation requires that the semantic part clusters do not overlap, which generally applies to our DSM.

For the prior distribution, if we expand it to the joint distribution of all the stroke exemplars, we obtain:

$$\begin{aligned} p(F|\theta) &= p(s_1, \ldots, s_n, l_1, \ldots, l_n|\theta) \\ &= p(s_1, \ldots, s_n|l_1, \ldots, l_n, \theta) p(l_1, \ldots, l_n|\theta). \end{aligned}$$

Using the same independence assumption as Equation (2), we get

$$p(F|\theta) \propto \prod_{i=1}^{n} p(s_i|l_i, u_i) p(l_1, \ldots, l_n, \theta).$$

Since assuming the DSM forms a tree structured prior distribution (Felzenszwalb and Huttenlocher 2005) we further obtain:

$$p(F|\theta) \propto \prod_{i=1}^{n} p(s_i|l_i, u_i) \prod_{(v_i, v_j) \in E} p(l_i, l_j|c_{ij}). \tag{3}$$

$p(s_i|l_i, u_i)$ is the probability of selecting stroke exemplar s_i from a semantic stroke cluster v_i, and it is constant once θ is obtained. So the final prior formulation is:

$$p(F|\theta) \propto \prod_{(v_i, v_j) \in E} p(l_i, l_j|c_{ij}). \tag{4}$$

Finally, using Eqs. (2) and (4), the posterior distribution of a configuration given an image can be written as:

$$p(F|I, \theta) \propto \prod_{i=1}^{n} p(I|s_i, l_i) \prod_{(v_i, v_j) \in E} p(l_i, l_j|c_{ij}). \tag{5}$$

where the first term encodes the fit to the image, and the second term encodes the plausibility of the geometric layout under the learned spatial prior.

4.2 Model Learning

The learning of a part-based model like DSM normally requires part-level supervision, however this supervision would be tedious to obtain for sketches. To substitute this part-level supervision, we propose a perceptual grouping algorithm to automatically segment sketches into semantic parts and employ a spectral clustering method (Zelnik-Manor and Perona 2004) to group these segmented semantic strokes into semantic stroke clusters. From the semantic stroke clusters, the model parameter θ will be learned through MLE.

4.2.1 Perceptual Grouping for Raw Strokes

Perceptual grouping creates the building blocks (*semantic strokes/parts*) for model learning based on *raw stroke* input. There are many factors that need to be considered in perceptual grouping. As demonstrated in Sect. 3, small strokes need to be grouped to be semantically meaningful, and local temporal order is helpful to decide whether strokes are semantically related. Equally important to the above, conventional perceptual grouping principles (Gestalt principles, e.g. proximity, continuity, similarity) are also required to decide if a stroke set should be grouped. Furthermore, after the first iteration, the learned DSM model is able to assign a group label for each stroke, which can be used in the next grouping iteration.

Algorithmically, our perceptual grouping approach is inspired by Barla et al. (2005), who iteratively and greedily group pairs of lines with minimum error. However, their cost function includes only proximity and continuity; and their purpose is line simplification, so grouped lines are replaced by new combined lines. We adopt the idea of iterative grouping but change and expand their error metric to suit our task. For grouped strokes, each stroke is still treated independently, but the stroke length is updated with the group length.

More specifically, for each pair of strokes s_1, s_2, grouping error is calculated based on 6 aspects: proximity, continuity, similarity, stroke length, local temporal order and model label (only used from second iteration), and the cost function is defined as:

$$\begin{aligned} Z(s_i, s_j) = (&\omega_{pro} * D_{pro}(s_i, s_j) + \omega_{con} * D_{con}(s_i, s_j) \\ &+ \omega_{len} * D_{len}(s_i, s_j) - \omega_{sim} * B_{sim}(s_i, s_j)) \\ &* J_{temp}(s_i, s_j) * J_{mod}(s_i, s_j), \end{aligned} \tag{6}$$

where proximity D_{pro}, continuity D_{con} and stroke length D_{len} are treated as cost/distance which increase the error, while similarity B_{sim} decreases the error. Local temporal order J_{temp} and model label J_{mod} further modulate the overall error. All the terms have corresponding weights $\{\omega\}$, which make the algorithm customizable for different datasets. Detailed definitions and explanations for the 6 terms follow below. Note that our perceptual grouping method is an unsupervised greedy algorithm, the colored perceptual grouping results (in Figs. 6, 7, 8, 9, 10) are just for differentiating grouped semantic strokes in individual sketches and have no correspondence between sketches.

Proximity Proximity employs the modified Hausdorff distance (MHD) (Dubuisson and Jain 1994) $d_H(\cdot)$ between two strokes, which represents the average closest distance

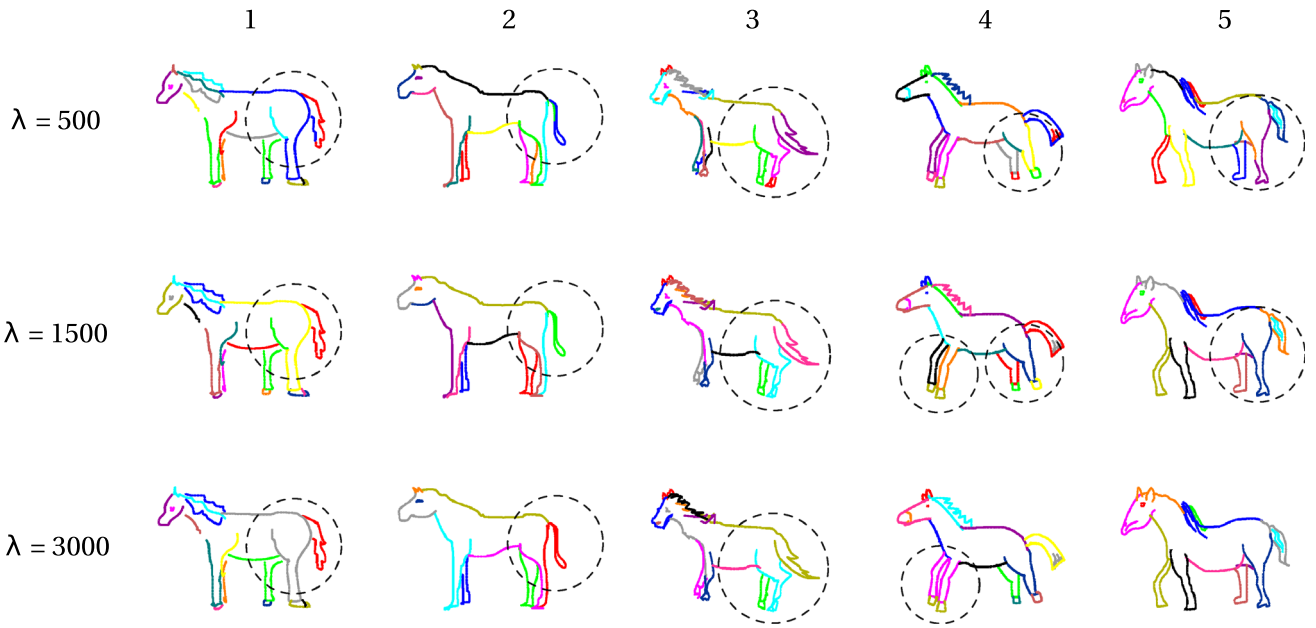

Fig. 6 The effect of changing λ to control the semantic stroke length (measured in pixels). We can see as λ increases, the semantic strokes' lengths increase as well.Generally speaking, when a proper semantic length is set, the groupings of the strokes are more semantically proper (neither over-segmented or over-grouped). More specifically, we can see that when $\lambda = 500$, many tails and back legs are fragmented. But when $\lambda = 1500$, those tails and back legs are grouped much better. Beyond that, when $\lambda = 3000$, two more semantic parts tend to be grouped together improperly, e.g., one back leg and the tail (*column 2*), the tail and the back (*column 3*), or two front legs (*column 4*). Yet it can also be noticed that when a horse is relatively well drawn (each part is very distinguishable), the stroke length term has less influence, e.g., column 5

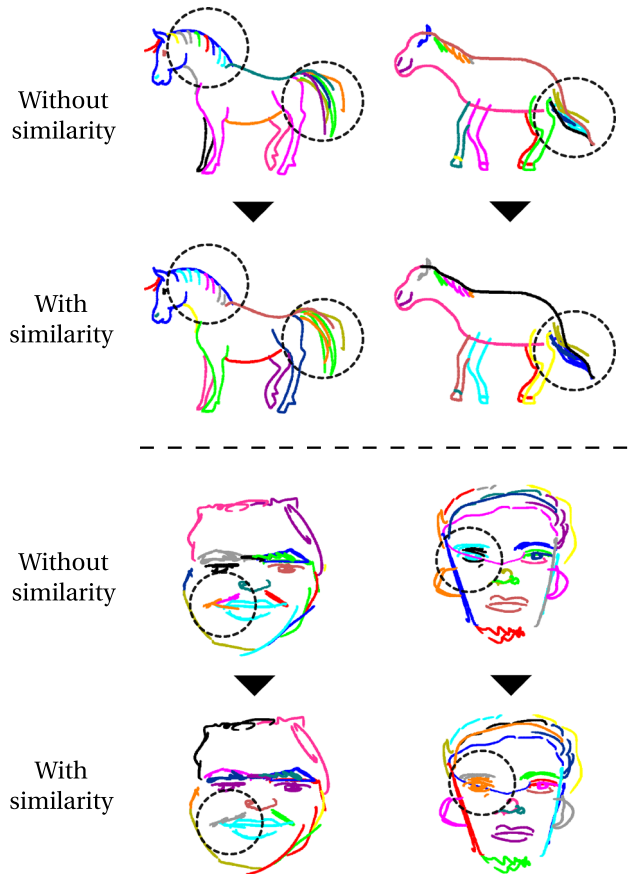

Fig. 7 The effect of the similarity term. Many separate strokes or wrongly grouped strokes are correctly grouped into properer semantic strokes when exploiting similarity

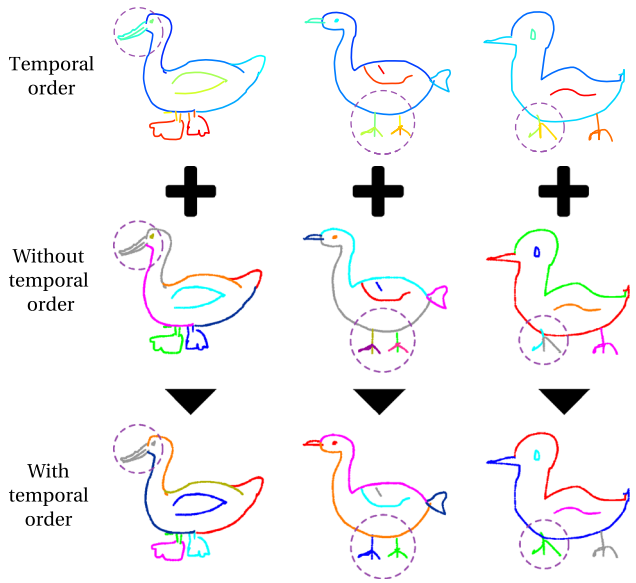

Fig. 8 The effect of employing stroke temporal order. It corrects many errors on the beak and feet (wrongly grouped with other semantic part or separated into several parts)

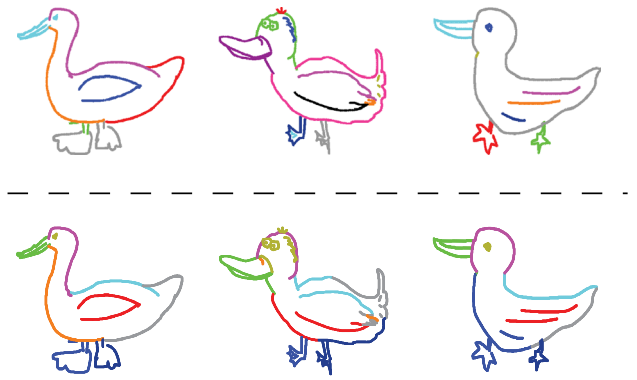

Fig. 9 The model label after the first iteration of perceptual grouping. *Above* first iteration perceptual groupings. *Below* model labels. It can be observed that the first iteration perceptual groupings have different number of semantic strokes, and the divisions over the eyes, head and body are quite different across sketches. However, after a category-level DSM is learned, the model labels the sketches in a very similar fashion, roughly dividing the duck into beak (*green*), head (*purple*), eyes (*gold*), back (*cyan*), tail (*grey*), wing (*red*), belly (*orange*), left foot (*light blue*), right foot (*dark blue*). But errors still exist in the model label, e.g., missing part or labeled part, which will be corrected in subsequent iterations

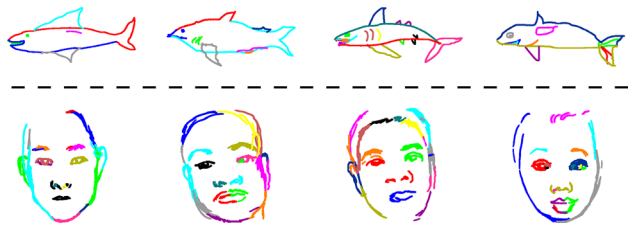

Fig. 10 Perceptual grouping results. For each sketch, a semantic stroke is represented by one color

between two sets of edge points. We define

$$D_{pro}(s_i, s_j) = d_H(s_i, s_j)/\epsilon_{pro},$$

dividing the calculated MHD with a factor ϵ_{pro} to control the scale of the expected proximity. Given the image size ϕ and the average semantic stroke number η_{avg} of the previous iteration (the average raw stroke number for the first iteration), we use $\epsilon_{pro} = \sqrt{\phi/\eta_{avg}}/2$, which roughly indicates how closely two semantically correlated strokes should be located.

Continuity To compute continuity, we first find the closest endpoints x,y of the two strokes. For the endpoints x,y, another two points x',y' on the corresponding strokes with very close distance (e.g., 10 pixels) to x,y are also extracted to compute the connection angle. Finally, the continuity is computed as:

$$D_{con}(s_i, s_j) = \|x - y\| * (1 + angle(\overrightarrow{x'x}, \overrightarrow{y'y}))/\epsilon_{con},$$

where ϵ_{con} is used for scaling, and set to $\epsilon_{pro}/4$, as continuity should have more strict requirement than the proximity.

Stroke Length Stroke length cost is the sum of the length of the two strokes:

$$D_{len}(s_i, s_j) = (P(s_i) + P(s_j))/\lambda, \tag{7}$$

where $P(s_i)$ is the length (pixel number) of raw stroke s_i; or if s_i is already within a grouped semantic stroke, it is the stroke group length. The normalization factor is computed as $\lambda = \tau * \eta_{sem}$, where η_{sem} is the estimated average number of strokes composing a semantic group in a dataset (from the analysis). When $\eta_{sem} = 1$, τ is the proper length for a stroke to be semantically meaningful (e.g. around 1500 px in Fig. 3a), and when $\eta_{sem} > 1$, τ is the maximum length of all the strokes.

The effect of changing λ to control the semantic stroke length is demonstrated in Fig. 6.

Similarity In some sketches, repetitive short strokes are used to draw texture like hair or mustache. Those strokes convey a complete semantic stroke, yet can be clustered into different groups by continuity. To correct this, we introduce a similarity bonus. We extract strokes s_1 and s_2's shape context descriptor and calculate their matching cost $K(s_i, s_j)$ according to Belongie et al. (2002). The similarity bonus is then:

$$B_{sim}(s_i, s_j) = exp(-K(s_i, s_j)^2/\sigma^2), \tag{8}$$

where σ is a scale factor. Examples in Fig. 7 demonstrate the effect of this term.

Local Temporal Order The local temporal order provides an adjustment factor J_{temp} to the previously computed error $Z(s_i, s_j)$ based on how close the drawing orders of the two strokes are:

$$J_{temp}(s_i, s_j) = \begin{cases} 1 - \mu_{temp}, & \text{if} |T(s_i) - T(s_j)| < \delta. \\ 1 + \mu_{temp}, & \text{otherwise.} \end{cases},$$

where $T(s)$ is the order number of stroke s. $\delta = \lceil \eta_{all}/\eta_{avg} \rceil$ is the estimated maximum order difference in stroke order within a semantic stroke, where η_{all} is the overall stroke number in the current sketch. μ_{temp} is the adjustment factor. The effect by this term is demonstrated in Fig. 8.

Model Label The DSM model label provides a second adjustment factor according to whether two strokes have the same label or not.

$$J_{mod}(s_i, s_j) = \begin{cases} 1 - \mu_{mod}, & \text{if} W(s_i) == W(s_j). \\ 1 + \mu_{mod}, & \text{otherwise.} \end{cases}, \tag{9}$$

where $W(s)$ is the model's label for stroke s, and μ_{mod} is the adjustment factor. The model label obtained after first iteration of perceptual grouping is shown in Fig. 9. Pseudo code for our perceptual grouping algorithm is shown in Algorithm 1. More results produced by first iteration perceptual grouping are illustrated in Fig. 10. As can be seen, every sketch is grouped into a similar number of parts, and there is reasonable group correspondence among the sketches in terms of appearance and geometry. However, obvious disagreement also can be observed, e.g., the tails of the sharks are grouped quite differently, as the same to the lips. This is due to the different ways of drawing one semantic stroke that are used by different sketches. This kind of intra-category semantic stroke variations are further addressed by our iterative learning scheme introduced in Sect. 4.4.

Algorithm 1 Perceptual grouping algorithm

```
Input t strokes {s_i}_{i=1}^t
Set the maximum error threshold to β
for i, j = 1 → t do
    ErrorMx(i, j) = Z(s_i, s_j)                ▷ Pairwise error matrix
end for
while 1 do
    [s_a, s_b, minError] = min(ErrorMx)        ▷ Find s_a, s_b with the
        smallest error
    if minError == β then
        break
    end if
    ErrorMx(a, b) ← β
    if None of s_a, s_b is grouped yet then
        Make a new group and group s_a, s_b
    else if One of s_a, s_b is not grouped yet then
        Group s_a, s_b to the existing group
    else
        continue
    end if
    Update ErrorMx cells that are related to strokes in the current
        group according to the new group length
end while
Assign each orphan stroke a unique group id
```

4.2.2 Spectral Clustering On Semantic Strokes

DSM learning is now based on the semantic strokes output by the perceptual grouping step. Putting the semantic strokes from all training sketches into one pool (we use the sketches of mirrored poses to increase the training sketch number and flip them to the same direction), we use spectral clustering (Zelnik-Manor and Perona 2004) to form category-level semantic stroke clusters. The spectral clustering has the con-

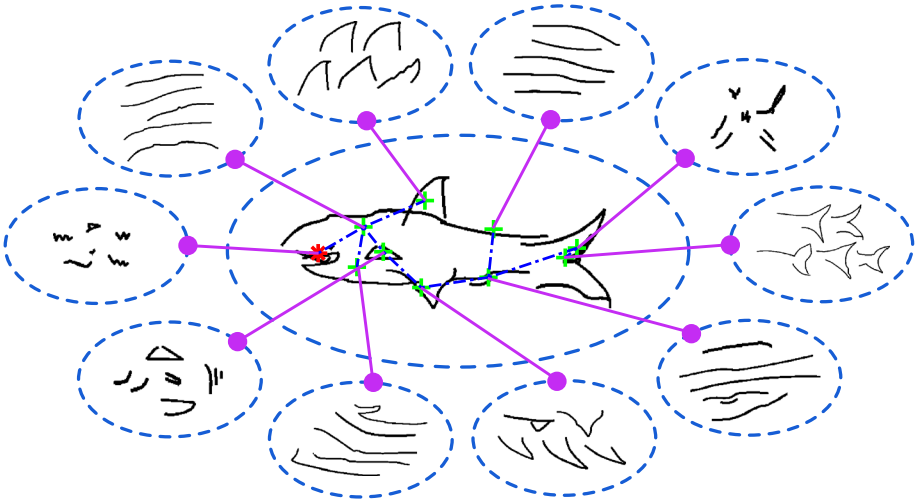

Fig. 11 An example of *shark* deformable stroke model with demonstration of the part exemplars in each semantic part cluster (*blue dashed ovals*), and the minimum spanning tree structure (*green crosses* for tree nodes and the *dash-dot lines* for edges)

venience of taking an arbitrary pairwise affinity matrix as input. Exploiting this, we define our own affinity measure A_{ij} for semantic strokes s_i, s_j whose geometrical centers are l_i, l_j as

$$A_{ij} = exp\left(\frac{-K(s_i, s_j)\|l_i - l_j\|}{\rho_{s_i}\rho_{s_j}}\right),$$

where $K(\cdot)$ is the shape context matching cost and ρ_{s_i} is the local scale at each stroke s_i (Zelnik-Manor and Perona 2004).

The number of clusters for each category is decided by the mean number of semantic strokes obtained by the perceptual grouper in each sketch. After spectral clustering, in each cluster, the semantic strokes generally agree on the appearance and location. Some cluster examples can be seen in Fig. 11.

Subsequently, unlike the conventional pictorial structure/deformable part-based model approach of learning parameters by optimizing on images, we follow contour model methods by learning model parameters from semantic stroke clusters.

Given $U_i = \{s_i^b\}_{b=1}^{M_i}$ representing the set of all strokes in semantic stroke cluster v_i and $L_i = \{l_i^b\}_{b=1}^{M_i}$ representing the geometrical centers of all M_i strokes in that cluster, the MLE estimate of θ is the value θ^* that maximizes $p(U_1, \ldots, U_n, L_1, \ldots, L_n|\theta)$.

$$\begin{aligned}\theta^* &= \arg\max_\theta p(U_1, \ldots, U_n, L_1, \ldots, L_n|\theta) \\ &= \arg\max_\theta p(U_1, \ldots, U_n|L_1, \ldots, L_n, \theta)p(L_1, \ldots, L_n|\theta).\end{aligned}$$

Similarly to Eq. (3), we have

$$\theta^* = \arg\max_\theta \prod_{i=1}^{n} p(U_i|L_i, u_i) \prod_{(v_i,v_j)\in E} p(L_i, L_j|c_{ij}). \quad (10)$$

Because the first term relies purely on the appearance of the strokes, and the second term relies purely on the cluster connectivity and the spatial relations between connected clusters, we can solve the two terms separately as described in the following sections.

4.2.3 Semantic Stroke Exemplar Learning

From Eq. (10), we can get the MLE estimate u^* for the appearance parameter u as:

$$u^* = \arg\max_u \prod_{i=1}^{n} p(U_i|L_i, u_i).$$

This is equivalent to independently solving for u_i^*:

$$u_i^* = \arg\max_{u_i} p(U_i|L_i, u_i).$$

Assuming each semantic stroke is generated independently, we obtain:

$$u_i^* = \arg\max_{u_i} \prod_{b=1}^{M_i} p(s_i^b|l_i^b, u_i), \quad (11)$$

where s_i^b and l_i^b are obtained directly from the semantic stroke cluster v_i, where we model

$$\begin{aligned}p(s_i^b|l_i^b, u_i) &= \arg\max_{s_i^a \in u_i} B_{sim}(s_i^b, s_i^a) \\ &= \arg\max_{s_i^a \in u_i} exp(-K(s_i^b, s_i^a)^2/\sigma^2),\end{aligned}$$

with Eq. (8). Therefore, Eq. (11) has no unique solution and depends on the strategy of selecting the stroke exemplars. Practically, we choose the m_i strokes with the lowest average shape context matching cost ($K(\cdot)$) to the others in each cluster v_i as the stroke exemplars $u_i = \{s_i^a\}_{a=1}^{m_i}$ (inspired by Shotton et al. (2008)). The exemplar number m_i is set to a

fraction of the overall stroke number in the obtained semantic stroke cluster v_i according to the quality of the training data, i.e., the better the quality, the bigger the fraction. Besides, we augment the stroke exemplars with their rotation variations to achieve more precise fitting. Some learned exemplar strokes of the *shark* category are shown in Fig. 11.

4.2.4 Spatial Relation Learning

From Equation (10), we get the MLE estimates E^* and c^* for the connectivity and the spatial relation parameters:

$$E^*, c^* = \arg\max_{E,c} \prod_{(v_i,v_j)\in E} p(L_i, L_j|c_{ij}).$$

Assuming each sketch is independently generated, we can further write

$$E^*, c^* = \arg\max_{E,c} \prod_{(v_i,v_j)\in E} \prod_{k=1}^{M_{ij}} p(l_i^k, l_j^k|c_{ij}), \tag{12}$$

where k indexes such stroke pairs that one stroke is from cluster v_i and the other from cluster v_j and they are from the same sketch.

Spatial Relations Before the MST structure is finalized, we can learn the spatial relation of each pair of connected clusters. To obtain relative location parameter c_{ij} for a given edge, we assume that offsets are normally distributed:

$$p(l_i^k, l_j^k|c_{ij}) = \mathcal{N}(l_i^k - l_j^k|\mu_{ij}, \Sigma_{ij}).$$

Then MLE result of:

$$(\mu_{ij}^*, \Sigma_{ij}^*) = \arg\max_{\mu_{ij}^*, \Sigma_{ij}^*} \prod_{k=1}^{M_{ij}} \mathcal{N}(l_i^k - l_j^k|\mu_{ij}, \Sigma_{ij}),$$

straightforwardly provides the estimate $c_{ij}^* = (\mu_{ij}^*, \Sigma_{ij}^*)$.

Learning the MST Structure To learn such an MST structure for E, we first define the quality of an edge (v_i, v_j) connecting two clusters with the MLE estimate c_{ij}^* as:

$$q(v_i, v_j) = \prod_{k=1}^{M_{ij}} p(l_i^k, l_j^k|c_{ij}^*).$$

Plugging this into Eq. (12), we obtain the MLE estimate E^* and convert the MLE into a minimization problem:

$$\begin{aligned} E^* &= \arg\max_E \prod_{(v_i,v_j)\in E} q(v_i, v_j) \\ &= \arg\min_E \sum_{(v_i,v_j)\in E} -\log q(v_i, v_j). \end{aligned}$$

Now solving for E^* is the same as obtaining the MST structure of the model graph G. This can be solved directly by the standard Kruskal's algorithm (Cormen et al. 2009).

The learned edge structure is illustrated in Figs. 1 and 11 by the green crosses and the blue dashed lines.

4.3 Model Detection

As discussed in Felzenszwalb and Huttenlocher (2005), matching DSM to sketches or images should include two steps: model configuration sampling and configuration energy minimization. Here, we employ fast directional chamfer matching (FDCM) (Liu et al. 2010) as the basic operation of stroke registration for these two steps, which is proved both efficient and robust at edge/stroke template matching (Thayananthan et al. 2003). In our framework, automatic sketch model detection is used in both iterative model training and image-sketch synthesis. This section explains this process.

4.3.1 Configuration Sampling

A configuration of the model $F = \{(s_i, l_i)\}_{i=1}^n$ is a model instance registered on an image. In one configuration, exactly one stroke exemplar s_i is selected in each cluster and placed at location l_i. Later, the configuration will be optimized by energy minimization to achieve best balance between (edge map) appearance and (model prior) geometry. Multiple configurations can be sampled, among which the best fitting can be chosen after energy minimization.

To achieve this, on a given image I and for the cluster v_i, we first sample possible locations for all the stroke exemplars $\{s_i^a\}_{a=1}^{m_i}$ with FDCM (one stroke exemplar may have multiple possible positions). A sampling region is set based on v_i's average bounding box to increase efficiency, and only positions within this region will be returned by FDCM. All the obtained stroke exemplars and corresponding locations form a set $H_m(v_i) = \{(s_i^z, l_i^z)\}_{z=1}^{h_i} (h_i \geq m_i)$. For each (s_i^z, l_i^z), a chamfer matching cost $D_{cham}(s_i^z, l_i^z, I)$ will also be returned, and only the matchings with a cost under a predefined threshold will be considered by us.

The posterior probability of a configuration F is described in Eq. (5). As the graph E forms a MST structure, each node is dependent on a parent node except the root node which is leading the whole tree. Letting v_r denote the root node, C_i denote child nodes of v_i, we can firstly sample a stroke exemplar and its location for the root according to the mar-

ginalized posterior probability $p(s_r, l_r|I, \theta)$, and then sample stroke exemplars and corresponding locations for its children $\{v_c|v_c \in C_r\}$ until we reach all the leaf nodes. The marginal distribution for the root can be written as:

$$p(s_r, l_r|I, \theta) \propto p(I|s_r, l_r) \prod_{v_c \in C_r} S_c(l_r),$$

$$S_j(l_i) \propto \sum_{(s_j, l_j) \in H_m(v_j)} \left(p(I|s_j, l_j) p(l_i, l_j|c_{ij}) \prod_{v_c \in C_j} S_c(l_j) \right).$$

And we define $p(I|s_i, l_i) = \exp(-D_{cham}(s_i, l_i, I))$.

In computation, the solution for the posterior probability of a configuration F is in a dynamic programming fashion. Firstly, all the S functions are computed once in a bottom-up order from the leaves to the root. Secondly, following a top-down order, we select the top f probabilities $p(s_r, l_r|I, \theta)$ for the root with corresponding f configurations $\{(s_r^b, l_r^b)\}_{b=1}^f$ for the root. For each root configuration (s_r^b, l_r^b), we then sample a configuration for its children that have the maximum marginal posterior probability:

$$p(s_j, l_j|l_i, I, \theta) \propto p(I|s_j, l_j) p(l_i, l_j|c_{ij}) \prod_{v_c \in C_j} S_c(l_j),$$

where i indexes the stroke exemplar from v_i the parent node and j indexes the stroke exemplar from v_j the child node. We continue this routine recursively until we reach the leaves. From this, we obtain f configurations $\{F_b\}_{b=1}^f$ for the model.

4.3.2 Energy Minimization

Energy minimization can be considered a refinement for a configuration F. It is solved similarly to configuration sampling with dynamic programming. But instead working with the posterior, it works with the energy function obtained by taking the negative logarithm (specifically natural logarithm for the convenience of computation) of Eq. (5):

$$L^* = \arg\min_L \left(\sum_{i=1}^n D_{cham}(s_i, l_i, I) + \sum_{(v_i, v_j) \in E} D_{def}(l_i, l_j) \right), \tag{13}$$

where $D_{def}(l_i, l_j) = -\ln p(l_i, l_j|c_{ij})$ is the deformation cost between each stroke exemplar and its parent exemplar, and $L = \{l_i\}_{i=1}^n$ are the locations for the selected stroke exemplars in F. The searching space for each l_i is also returned by FDCM. Comparing to configuration sampling, we set a higher threshold for FDCM, and for each stroke exemplar s_i in F, a new series of locations $\{(s_i, l_i^k)\}$ are returned by FDCM. A new l_i is then chosen from those candidate locations $\{l_i^k\}$. To make this solvable by dynamic programming, we define:

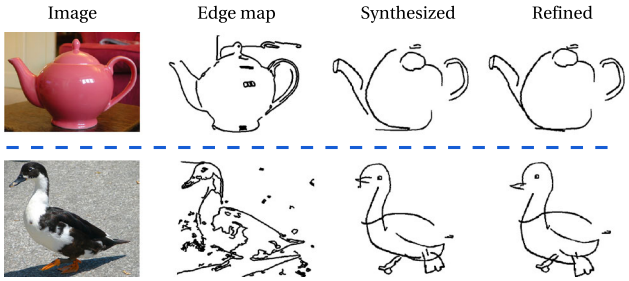

Fig. 12 Refinement results illustration

$$Q_j(l_i) = \min_{l_j \in \{l_j^k\}} (D_{cham}(s_j, l_j, I) + D_{def}(l_i, l_j) + \sum_{v_c \in C_j} Q_c(l_j)), \tag{14}$$

By combining Eqs. (13) and (14) and exploit the MST structure again, we can formalize the energy objective function of the root node as:

$$l_r^* = \arg\min_{l_r \in \{l_r^k\}} \left(D_{cham}(s_r, l_r, I) + \sum_{v_c \in C_r} Q_c(l_j) \right).$$

Through the same bottom-up routine to calculate all the Q functions and the same top-down routine to find the best locations from the root to the leaves, we can find the best locations L^* for all the exemplars. As mentioned before, we sampled multiple configurations and each will have a cost after energy minimization. We choose the one with lowest cost as our final detection result.

Aesthetic Refinement The obtained detection results sometimes will have unreasonable placement for the stroke exemplar due to the edge noise. To correct this kind of error, we perform another round of energy minimization, with appearance terms D_{cham} switched off. Rather than use chamfer matching to select the locations, we let the stroke exemplar to shift around its detection position within a quite small region. Some refinement results are shown for the image-sketch synthesis process in Fig. 12.

4.4 Iterative Learning

As stated before, the model learned with one pass through the described pipeline is not satisfactory—with duplicated and missing semantic strokes. To improve the quality of the model, we introduce an iterative process of: (1) perceptual grouping, (2) model learning and (3) model detection on training data in turns. The learned model will assign cluster labels for raw strokes during detection according to which stroke exemplar the raw stroke overlaps the most with or has the closest distance to. And the model labels are used in

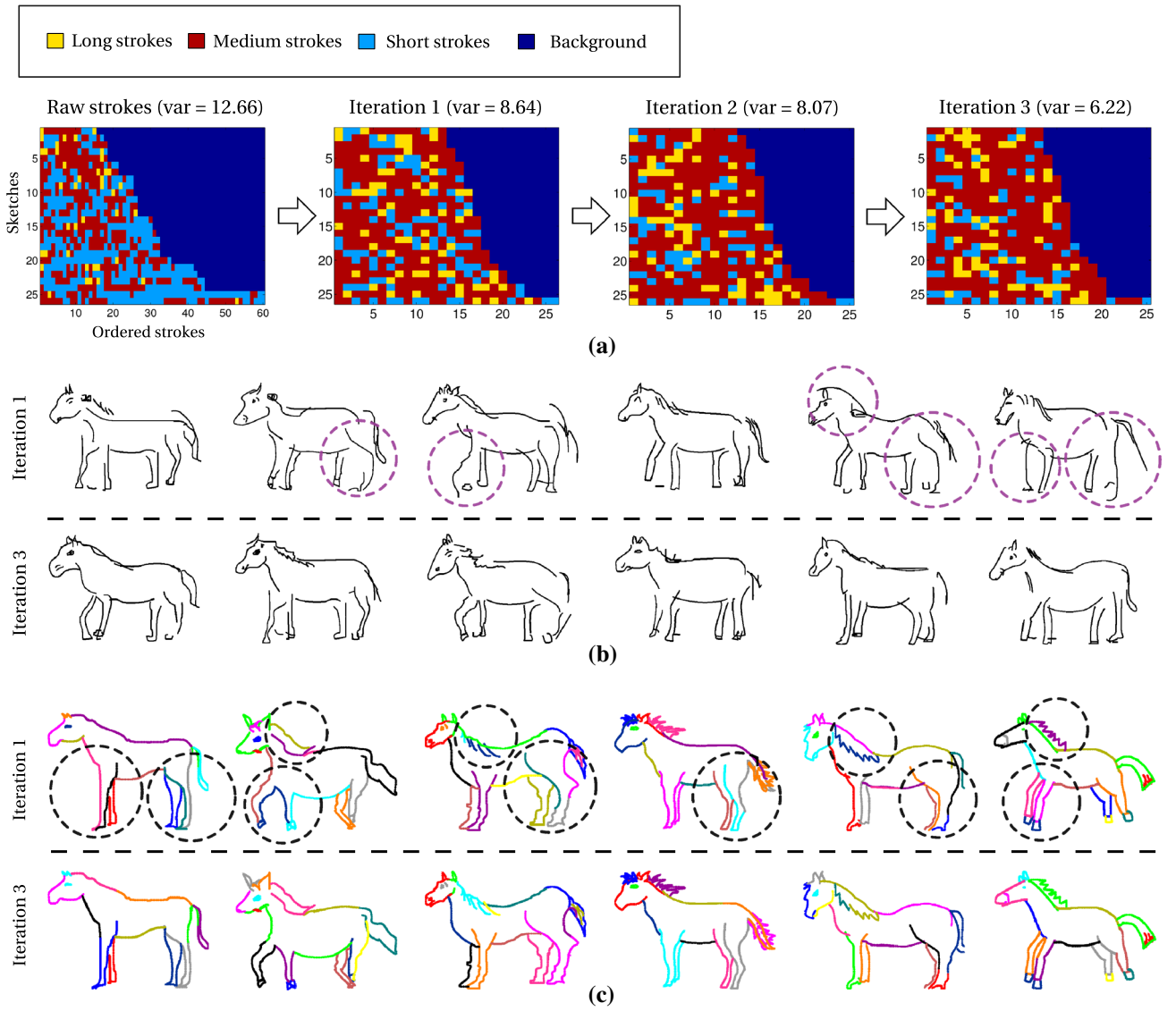

Fig. 13 The convergence process during model training (horse category): **a** semantic stroke number converging process (*var* denotes variance); **b** learned horse models at iteration 1 and 3 (We pick one stroke exemplar from every stroke cluster each time to construct a horse model instance, totally 6 stroke exemplars being chosen and resulting 6 horse model instances); **c** Perceptual grouping results at iteration 1 and 3. Comparing to iteration 1, a much better consensus on the legs and the neck of the horse is observed on iteration 3 (flaws in iteration 1 are highlighted with *dashed circles*). This is due to the increased quality of the model of iteration 3, especially on the legs and the neck parts

the perceptual grouping in the next iteration (Eq. (9)). If an overly-long stroke crosses several stroke exemplars, it will be cut into several strokes to fit the corresponding stroke exemplars.

We employ the variance of semantic stroke numbers at each iteration as convergence metric. Over iterations, the variance decreases gradually, and we choose the semantic strokes from the iteration with the smallest variance to train the final DSM. Fig. 13a demonstrates the convergence process of the semantic stroke numbers during the model training. Different from Fig. 4, we use 3 colors here to represent the short strokes (cyan), medium strokes (red) and long strokes (yellow). As can be seen in the figure, accompanying the convergence of stroke number variance, strokes are formed into medium strokes with properer semantics as well. Fig. 13b illustrates the evolution of the stroke model during the training, and Fig. 13c shows the evolution of the perceptual grouping results.

4.5 Image-Sketch Synthesis

After the final DSM is obtained from the iterative learning, it can directly be used for image-sketch synthesis through model detection on an image edge map—where we avoid the localization challenge by assuming an approximate object bounding box has been given. Also the correct DSM (cat-

egory) has to be selected in advance. These are quite easy annotations to provide in practice.

5 Experiments

We evaluate our sketch synthesis framework (1) qualitatively by way of showing synthesized results, and (2) quantitatively via two user studies. We show that our system is able to generate output resembling the input image in plausible free-hand sketch style; and that it works for a number of object categories exhibiting diverse appearance and structural variations.

We conduct experiments on 2 different datasets: (1) TU-Berlin, and (2) Disney portrait. TU-Berlin dataset is composed of non-expert sketches while Disney portrait dataset is drawn by selected professionals. 10 testing images of each category are obtained from ImageNet, except the face category where we follow Berger et al. (2013) to use the Center for Vital Longevity Face Database (Minear and Park 2004). To fully use the training data of the Disney portrait dataset, we did not synthesize face category using images corresponding to training sketches of Disney portrait dataset, but instead selected 10 new testing images to synthesize from. We normalize the grayscale range of the original sketches to 0 to 1 to simplify the model learning process. Specifically, we chose 6 diverse categories from TU-Berlin: *horse*, *shark*, *duck*, *bicycle*, *teapot* and *face*; and the 90*s* and 30*s* abstraction level sketches from *artist A* and *artist E* from Disney portrait (270 level is excluded considering the high computational cost and 15*s* level is due to the presence of many incomplete sketches).

5.1 Free-Hand Sketch Synthesis Demonstration

In Fig. 14, we illustrate synthesis results for five categories using models trained on the TU-Berlin dataset. We can see that synthesized sketches resemble the input images, but are clearly of free-hand style and abstraction. In particular, (1) major semantic strokes are respected in all synthesized sketches, i.e., there are no missing or duplicated major semantic strokes, (2) changes in intra-category body configurations are accounted for, e.g., different leg configurations of horses, and (3) part differences of individual objects are successfully synthesized, e.g., different styles of feet for duck and different body curves of teapots.

Fig. 15 offers synthesis results for *face* only, with a comparison between these trained on the TU-Berlin dataset and Disney portrait dataset. In addition to the above observations, it can be seen that when professional datasets (e.g., portrait sketches) are used, synthesized faces tend to be more precise and resemble better the input photo. Furthermore, when compared with Berger et al. (2013), we can see that although without intense supervision (the fitting of a face-specific mesh model), our model still depicts major facial components with reasonable precision and plausibility (except for hair which is too diverse to model well), and yields similar synthesized results especially towards higher abstraction levels (Please refer to Berger et al. (2013) for result comparison). We acknowledge that the focus of Berger et al. (2013) is different to ours, and believe adapting detailed category-specific model alignment supervision could further improve the aesthetic quality of our results, especially towards the less abstract levels.

5.2 Perceptual Study

Two separate user studies were performed to quantitatively evaluate our synthesis results. We employed 10 different participants for each study (to avoid prior knowledge), making a total of 20. The first user study is on sketch recognition, in which humans are asked to recognize synthesized sketches. This study confirms that our synthesized sketches are semantic enough to be recognizable by humans. The second study is on perceptual similarity rating, where subjects are asked to link the synthesized sketches to their corresponding images. By doing this, we demonstrate the intra-category discrimination power of our synthesized sketches.

5.2.1 Sketch Recognition

Sketches synthesized using models trained on TU-Berlin dataset are used in this study, so that *human* recognition performance reported in Eitz et al. (2012) can be used as comparison. There are 60 synthesized sketches in total, with 10 per category. We equally assign 6 sketches (one from each category) to every participant and ask them to select an object category for each sketch (250 categories are provided in a similar scheme as in Eitz et al. (2012), thus chance is 0.4 %). From Table 1, we can observe that our synthesized sketches can be clearly recognized by humans, in some cases offering 100 % accuracy. We note that human recognition performance on our sketches follows a very similar trend across categories to that reported in Eitz et al. (2012). The overall higher performance of ours is most likely due to the much smaller scale of our study. The result of this study clearly shows that our synthesized sketches convey enough semantic meaning and are highly recognizable as human-drawn sketches.

5.2.2 Image-Sketch Similarity

For the second study, both TU-Berlin dataset and Disney portrait dataset are used. In addition to the 6 models from TU-Berlin, we also included 4 models learned using the 90*s* and 30*s* level sketches from *artist A* and *artist E* from Disney portrait dataset. For each category, we randomly chose

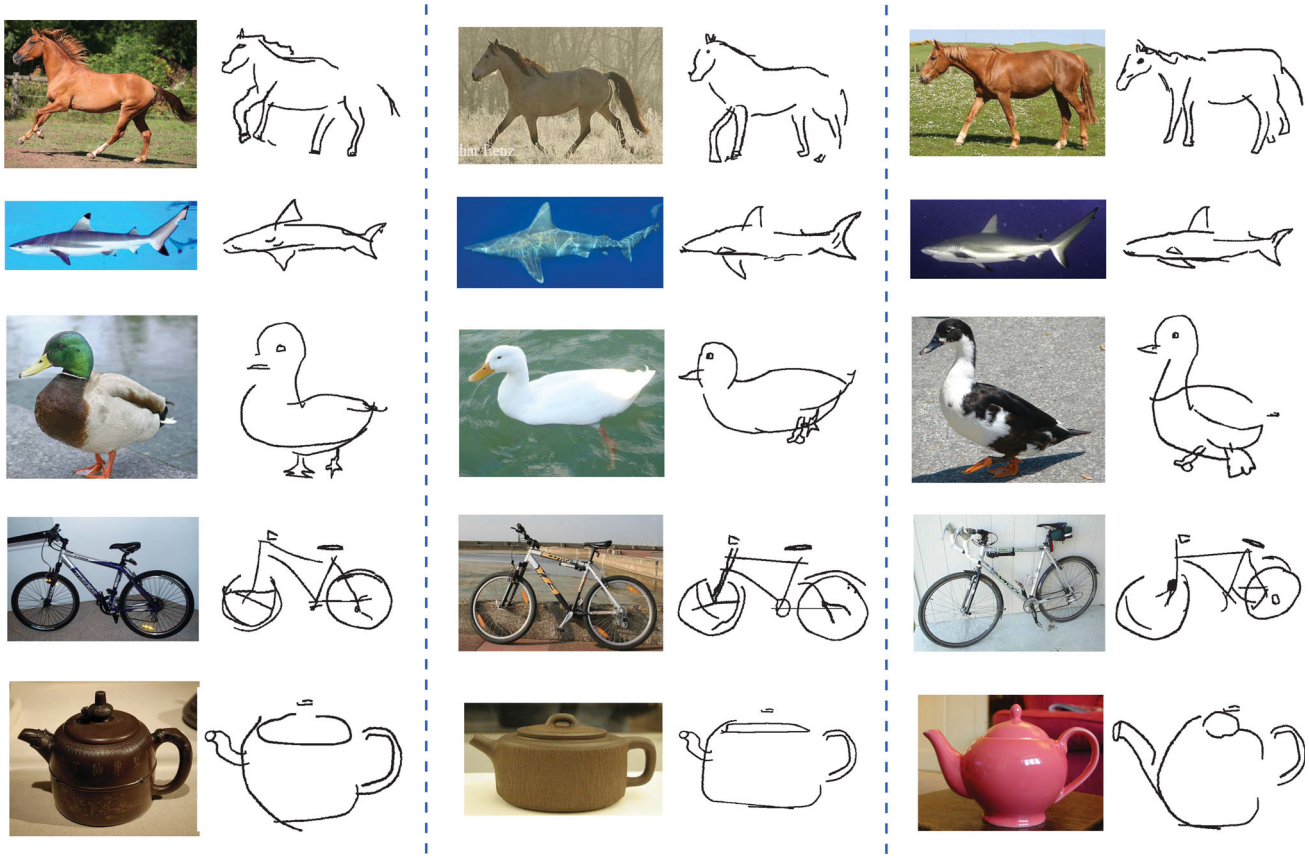

Fig. 14 Sketch synthesis results of five categories in the TU-Berlin dataset

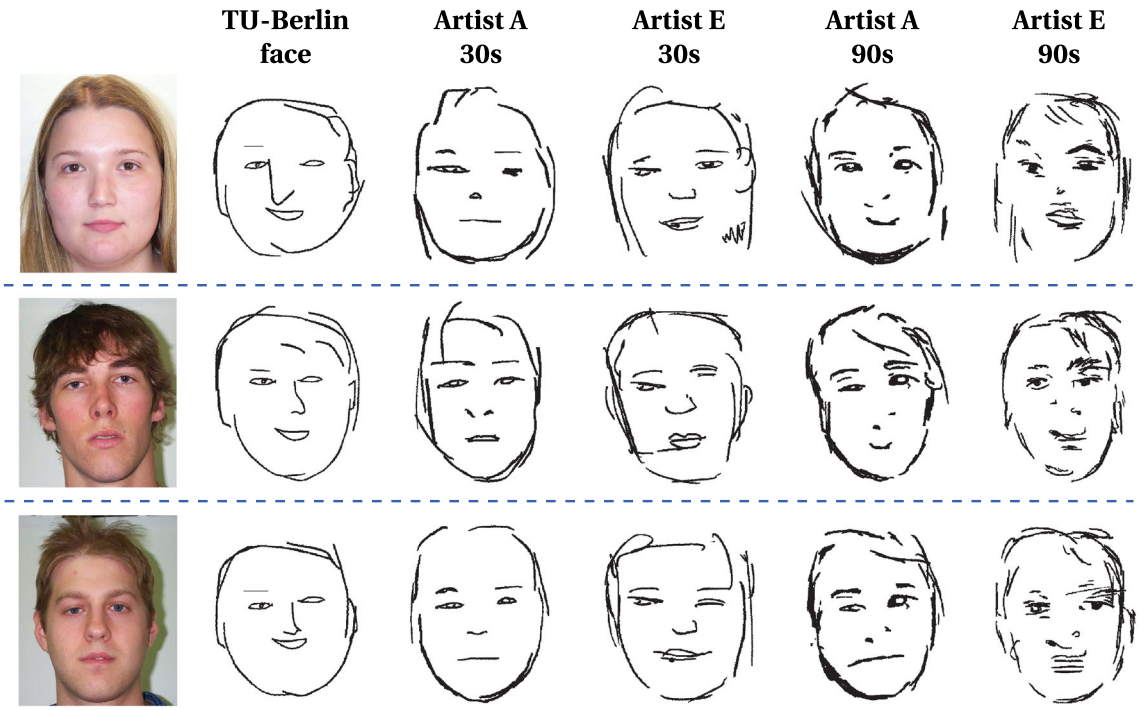

Fig. 15 A comparison of sketch synthesis results of *face* category using the TU-Berlin dataset and Disney portrait dataset

Table 1 Recognition rates of human users for (S)ynthesised and (R)eal sketches (Eitz et al. 2012)

	Horse (%)	Shark (%)	Duck (%)	Bicycle (%)	Teapot (%)	Face (%)
S	100	40	100	100	90	80
R	86.25	60	78.75	95	88.75	73.75

Table 2 Image-sketch similarity rating experiment results

	Horse	Shark	Duck	Bicycle	Teapot
Acc.	86.67 %	73.33 %	63.33 %	83.33 %	66.67 %
p	<0.01	<0.01	0.10	<0.01	<0.05
	Face	A 30s	E 30s	A 90s	E 90s
Acc.	76.67 %	76.67 %	90.00 %	73.33 %	56.67 %
p	<0.01	<0.01	<0.01	<0.01	0.29

3 image pairs, making 30 pairs (3 pairs × 10 categories) in total for each participant. Each time, we show the participant one pair of images and their corresponding synthesized sketches, where the order of sketches may be the same or reversed as the image order (Due to the high abstraction nature of the sketches, only a pair of sketches is used and two corresponding images are provided for clues each time). Please refer to Fig. 14 to see some example image and sketch pairs. The participant is then asked to decide if the sketches are of the same order as the images. We consider a choice to be correct if the participant correctly identified the right ordering. Finally, the accuracy for each category is averaged over 30 pairs and summarized in Table 2. A binomial test is applied to the results, and we can see that, except *duck* and *Artist E 90s*, all the rest results are significantly better than random guess (50 %), with most $p < 0.01$. The relatively weaker performance for *duck* and *teapot* from TU-Berlin is mainly due to a lack of training sketch variations as opposed to image domain, resulting in the model failing to capture enough appearance variations in images. On Disney portrait dataset, matching accuracy is generally on the same level as TU-Berlin, yet there appears to be a big divide on *artist E 90s*. This is self-explanatory when one compares synthesized sketches of the 90s level from *artist E* (last column of Fig. 15) with other columns—*artist E 90s* seems to depict a lot more short and detailed strokes making the final result relatively messy. In total, we can see that our synthesized sketches possess sufficient intra-category discrimination power.

5.3 Comparison With Other Works

5.3.1 Qualitative Comparision

To demonstrate the distinct free-hand style conveyed by our DSM, we select 4 major works that can also generate sketch-like images but employ different strategies as comparison: XDoG (Winnemöller 2011), FDoG (Kang et al. 2007), active basis model (ABM) (Wu et al. 2010) and sketch tokens (Lim et al. 2013). We use the available implementation for each method, and tune the parameter(s) moderately to generate as clear results as possible. The generated sketch-like images are demonstrated in Fig. 16. The NPAR works, i.e. XDoG and FDoG, have largely kept both foreground and background photo details. Although the aesthetics of the NPAR result is quite good when the textures of the background and foreground are not too complicated, only moderate abstraction is expressed in the results. Moreover, it would be hard to remove artifacts resulting from complicated texture. The ABM offers more abstract results and without background content, thus better simulating human sketching. However, due to the use of Gabor wavelets, constituent strokes (wavelets) are not similar to natural human strokes, and the level of detail is quite sparse. Sketch tokens method provides the closest results to human sketches except for our DSM results. They possess decent level of abstraction and depict enough details. However background artifacts are not totally avoided and little free-hand style is present. Uniquely, on the task of free-hand sketch synthesis, our DSM can generate sketch images that have good balance between abstraction and object detail and highly resemble the style of real free-hand sketches.

5.3.2 Sketch Recognition via SVM

We also try to evaluate the quality of each work quantitatively here through support vector machine (SVM) sketch recognition (Eitz et al. 2012). In Table 3, we compared the human sketch recognition rates for real sketches and those synthesized by our DSM. Here, we employ the SVM classifiers trained on the TU-Berlin sketch dataset to recognize the sketch-like images generated by different methods. The SVM recognition rates for different categories reported in Eitz et al. (2012) are also included as reference. The intuition here is that if the generated images highly resemble the free-hand sketch style and clearly depict the object in the image, they should be successfully recognized by the classifiers trained on a large-scale sketch dataset. We offer the recognition performances in Table 3. The results show that the sketches generated by DSM can be well recognized by the SVM classfiers and the recognition rates on DSM sketches of different categories are generally higher than the recogni-

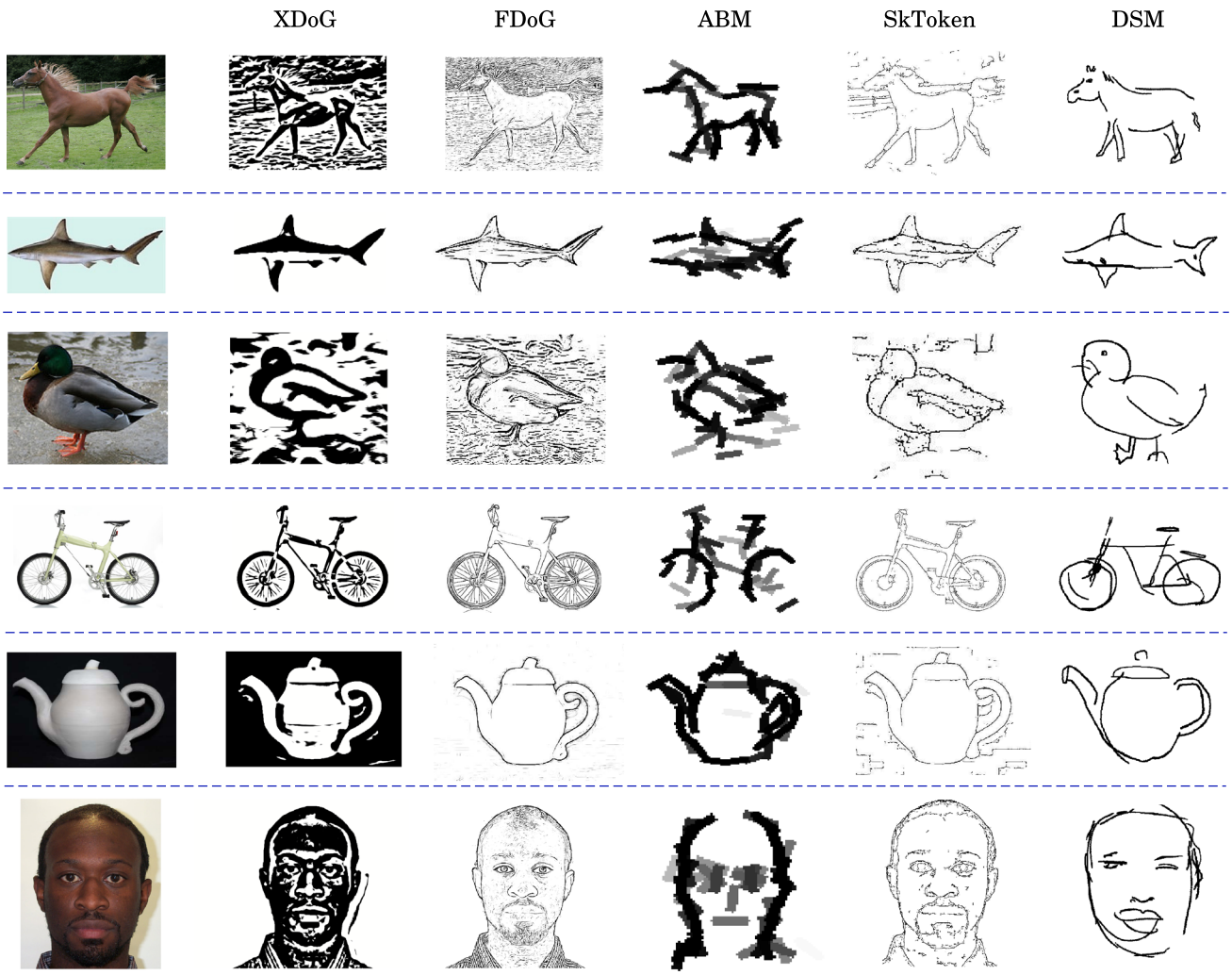

Fig. 16 Comparison of our DSM to 4 representative works which could also generate sketch-like results, including XDoG (Winnemöller 2011), FDoG (Kang et al. 2007), active basis model (ABM) (Wu et al. 2010) and sketch tokens (SkToken) (Lim et al. 2013)

Table 3 Sketch category recognition by SVM classifier. We compare recognition rates for sketches synthesised by DSM (ours), XDoG, FDoG, active basis model (ABM), sketch token (SkToken) and with the results reported in Eitz et al. (2012)

	Horse (%)	Shark (%)	Duck (%)	Bicycle (%)	Teapot (%)	Face (%)
XDoG	0	0	0	0	0	0
FDoG	0	0	0	30	0	0
ABM	0	0	0	0	0	0
SkToken	10	0	0	50	0	0
DSM	90	100	30	100	100	90
Real	53.85	65.39	48.15	76.92	70.37	44.44

tion rates reported in Eitz et al. (2012). We attribute this to the fact that our framework produces a normalized model for each sketch category, which could synthesize sketches that have slightly lower intra-class diversity than real sketches. For other methods, only the sketch token results on horse and bicycle categories, and FDoG results on bicycle category, could be recognized by the SVM classifiers. This is understandable given Fig. 16, as the results generated by the alternative methods either do not have clear free-hand style, miss some details or include too much noise. We admit that there is a bias using free-hand sketch classifiers, as the alternative results are sufficiently meaningful for human recognition. Nevertheless, our results have conveyed the clearest semantic meaning in the free-hand sketch format.

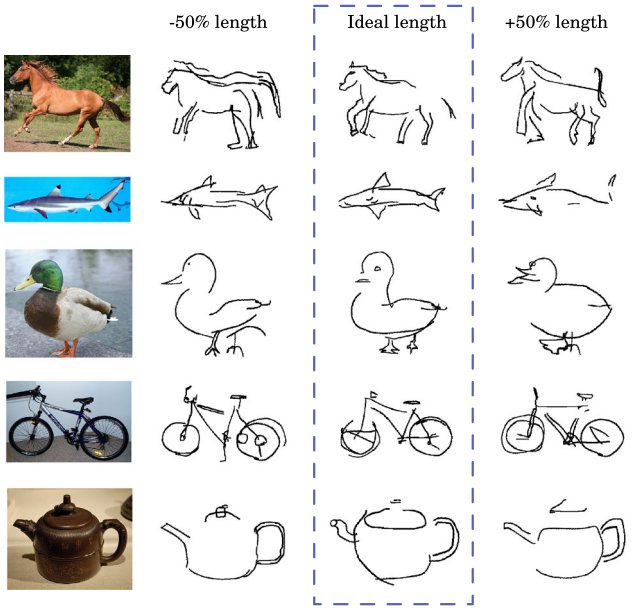

Fig. 17 Synthesis results from models trained with different semantic length priors: the ideal length, 50 % shorter (*left* −50 % length) and 50 % longer (*right* +50 % length) than the ideal length

5.4 Parameter Tuning and Failure Cases

Our model is intuitive to tune, with important parameters constrained within perceptual grouping. There are two sets of parameters affecting model quality: semantic stroke length and weights for different terms in Eq. (6). Semantic stroke length reflects negatively to the semantic stroke number and it needs to be tuned consistent with the statistical observation of that category. It is estimated as the λ illustrated in Eq. (7). For η_{sem} we used 1–3 for TU-Berlin dataset and the 30*s* level portrait sketches, and for the 90*s* level portrait sketches, η_{sem} is set 8 and 11 respectively for the *90s* level of *artist A* and *artist E*. This is because in the less abstracted sketches artists tend to use more short strokes to form one semantic stroke. For those categories with $\eta_{sem} = 1$, we found 85–95 % of the maximum stroke length is a good range to tune against for τ since our earlier stroke-level study suggests semantic strokes tend to cluster within this range (see Fig. 3). In Fig. 17, we demonstrate the effect of using different semantic length parameters. Besides DSMs trained with the proper stroke length setting, we also train DSMs with stroke lengths 50 % shorter and 50 % longer than the ideal length. The synthesis results obtained with these models are illustrated in Fig. 17. We observe that models trained with a shorter length prior tend to have duplicated parts and blurry synthesis results; while models trained with a longer length prior tend to be missing some parts and incomplete synthesis results. In both atypical cases, the model quality is downgraded as the parts are improperly formed.

Regarding weights for different terms in Eq. (6), we used the same parameters for both the TU-Berlin dataset and 30*s* level portrait sketches, and set ω_{pro}, ω_{con} and ω_{len} (for proximity, continuity and stroke length respectively) uniformly to 0.33. For the 90*s* level sketches, again since too many short strokes are used, we switched off the continuity term, and set ω_{pro} and ω_{len} both to 0.5. The weight ω_{sim} and adjustment factors μ_{temp} and μ_{mod} (corresponding to similarity, local temporal order and model label) are all fixed to 0.33 in all experiments.

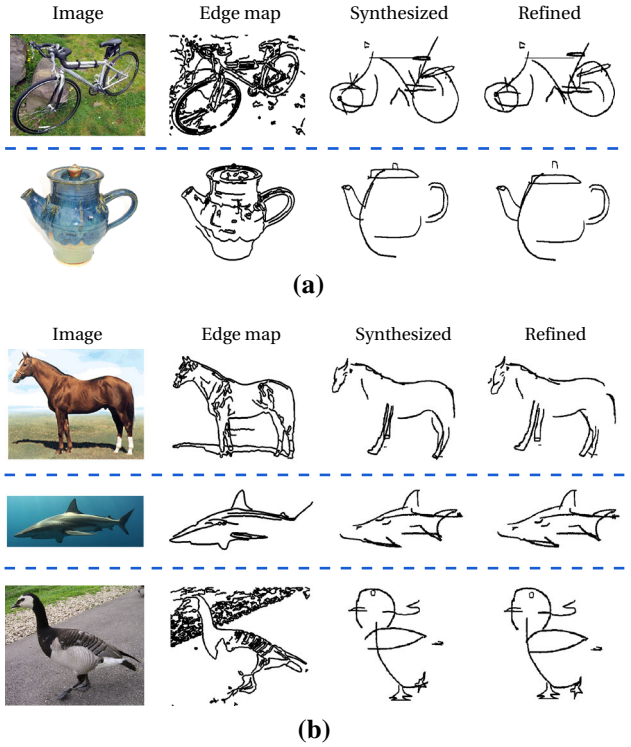

Fig. 18 Failures cases due to: **a** appearance or configuration variation is outside the model's learned distribution. **b** Severe edge noise or incomplete edge map

In Fig. 18, we show some failure examples and there are two major sources of failure. First, the given image object has some appearance or part configuration that is beyond our learned model's distribution. The second is severely noisy or incomplete edge maps.

6 Further Discussions

Data Alignment Although our model can address a good amount of variation in the number, appearance and location of parts without the need for well-aligned datasets, a poor model may be learned if the topology diversity (existence, number and layout of parts) of the training sketches is too extreme. This could be alleviated by selecting fine-grained sub-categories of sketches to train on, which would require more constrained collection of training sketches.

Model Quality Due to the unsupervised nature of our model, it has difficulty modelling challenging objects with complex inner structure. For example, buses often exhibit complicated features such as the number and location of windows. We expect that some simple user interaction akin to that used in interactive image segmentation could help to increase model precision, for example by asking the user to scribble an outline to indicate rough object parts.

Another weakness of our model is that the diversity of synthesized results is highly dependent on training data. If there are no similar sketches in the training data that can roughly resemble the input image, it will be hard to generate a good looking free-hand sketch for that image, e.g., some special shaped teapot images. We also share the common drawback of part-based models, that severe noise will affect detection accuracy.

Aesthetic Quality In essence, our model learns a normalized representation for a given category. However, apart from common semantic strokes, some individual sketches will exhibit unique parts not shared by others, e.g., saddle of a horse. To explicitly model those accessory parts can significantly increase the descriptive power of the stroke model, and thus is an interesting direction to explore in the future. Last but not least, as the main aim of this work is to tackle the modeling for category-agnostic sketch synthesis, only very basic aesthetic refinement post-processing was employed. A direct extension of current work will therefore be leveraging advanced rendering techniques from the NPR domain to further enhance the aesthetic quality of our synthesized sketches.

7 Conclusion

We presented a free-hand sketch synthesis system that for the first time works outside of just one object category. Our model is data-driven and uses publicly available sketch datasets regardless of whether drawn by non-experts or professionals. With minimum supervision, i.e., the user selects a few sketches of similar poses from one category, our model automatically discovers common semantic parts of that category, as well as encoding structural and appearance variations of those parts. Importantly, corresponding pairs of photo and sketch images are not required for training, nor any alignment is required. By fitting our model to an input image, we automatically generate a free-hand sketch that shares close resemblance to that image. Results provided in the previous section confirms the efficacy of our model.

References

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 898–916.

Barla, P., Thollot, J., Sillion, FX. (2005). Geometric clustering for line drawing simplification. In: Proceedings of the Eurographics Symposium on Rendering, pp. 183–192

Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(4), 509–522.

Berger, I., Shamir, A., Mahler, M., Carter, E., & Hodgins, J. (2013). Style and abstraction in portrait sketching. *ACM Trans Graph (Proc SIGGRAPH)*, *32*(4), 55:1–55:12.

Breiman, L. (2001). Random forests. Maching. *Learning*, *45*(1), 5–32.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–698.

Chen, H., Zheng, N., Liang, L., Li, Y., Xu, Y., Shum, H. (2002). Pictoon: A personalized image-based cartoon system. In: Proceedings of the Tenth ACM International Conference on Multimedia, pp. 171–178

Chen, H., Xu, Z., Liu, Z., Zhu, SC. (2006). Composite templates for cloth modeling and sketching. In: CVPR, pp. 943–950

Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H. S., Finkelstein, A., Funkhouser, T., et al. (2008). Where do people draw lines? *ACM Transactions on Graphics (Proc SIGGRAPH)*, *27*(3), 88:1–88:11.

Cormen, TH., Leiserson, CE., Rivest, RL., Stein, C. (2009). MIT Press

Dai, J., Wu, Y., Zhou, J., Zhu, S. (2013) Cosegmentation and cosketch by unsupervised learning. In: ICCV, pp. 1305–1312

DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., & Santella, A. (2003). Suggestive contours for conveying shape. *ACM Trans Graph (Proc SIGGRAPH)*, *22*(3), 848–855.

Dubuisson, MP., Jain, AK.: (1994). A modified hausdorff distance for object matching. In: International Conference on Patter Recognition, pp. 566–568

Eitz, M., Hays, J., & Alexa, M. (2012). How do humans sketch objects? *ACM Trans Graph (Proc SIGGRAPH)*, *31*(4), 44:1–44:10.

Everingham, M., Van Gool, L., Williams, CKI., Winn, J., Zisserman, A.: (2007) The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1), 55–79.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

Ferrari, V., Jurie, F., & Schmid, C. (2010). From images to shape models for object detection. *International Journal of Computer Vision*, *87*(3), 284–303.

Fu, H., Zhou, S., Liu, L., & Mitra, N. J. (2011). Animated construction of line drawings. *ACM Trans Graph (Proc SIGGRAPH)*, *30*(6), 133.

Gooch, B., Reinhard, E., & Gooch, A. (2004). Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans Graph (Proc SIGGRAPH)*, *23*(1), 27–44.

Grabli, S., Turquin, E., Durand, F., & Sillion, F. X. (2010). Programmable rendering of line drawing from 3d scenes. ACM Transactions on Graphics, 29(2), 18:1–18:20.

Guo, C., Zhu, S., & Wu, Y. (2007). Primal sketch: Integrating structure and texture. *Computer Vision and Image Understanding*, *106*(1), 5–19.

Huang, Z., Fu, H., & Lau, R. W. H. (2014). Data-driven segmentation and labeling of freehand sketches. *ACM Transactions on Graphics (Proc SIGGRAPH)*, *33*(6), 175:1–175:10.

Judd, T., Durand, F., & Adelson, E. H. (2007). Apparent ridges for line drawing. *ACM Transactions on Graphics*, *26*(3), 19.

Kang, H., Lee, S., Chui, CK. (2007). Coherent line drawing. In: Proceedings of the International Symposium on Non-photorealistic Animation and Rendering, pp. 43–50

Koffka, K., (1935). Principles of Gestalt Psychology

Kyprianidis, JE., Döllner, J. (2008). Image abstraction by structure adaptive filtering. In: Proc. EG UK Theory and Practice of Computer Graphics, pp. 51—58

Liang, L., Chen, H., Xu, Y., Shum, H. (2002). Example-based caricature generation with exaggeration. In: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, pp. 386–393

Lim, JJ., Zitnick, CL., Dollar, P. (2013). Sketch tokens: A learned mid-level representation for contour and object detection. In: CVPR, pp. 3158–3165

Liu, J., Fu, H., Tai, CL. (2014). Dynamic sketching: Simulating the process of observational drawing. In: Proceedings of the Workshop on Computational Aesthetics, pp. 15–22

Liu, M., Tuzel, O., Veeraraghavan, A., Chellappa, R. (2010). Fast directional chamfer matching. In: CVPR, pp. 1696–1703

Minear, M., & Park, D. (2004). A lifespan database of adult facial stimuli. *Behavior Research Methods, Instruments, & Computers*, *36*(4), 630–633.

Opelt, A., Pinz, A., Zisserman, A. (2006). A boundary-fragment-model for object detection. In: ECCV, pp. 575–588

Qi, Y., Guo, J., Li, Y., Zhang, H., Xiang, T., Song, Y. (2013). Sketching by perceptual grouping. In: ICIP, pp. 270–274

Qi, Y., Song, YZ., Xiang, T., Zhang, H., Hospedales, T., Li, Y., Guo, J. (2015). Making better use of edges via perceptual grouping. In: CVPR

Ren, X., Bo, L. (2012). Discriminatively trained sparse code gradients for contour detection. In: Advances in Neural Information Processing Systems, pp. 584–592

Ren, X., Fowlkes, C. C., & Malik, J. (2008). Learning probabilistic models for contour completion in natural images. *International Journal of Computer Vision*, *77*(1–3), 47–63.

Saund, E. (1992). Putting knowledge into a visual shape representation. *Artificial Intelligence*, *54*(1–2), 71–119.

Schneider, R. G., & Tuytelaars, T. (2014). Sketch classification and classification-driven analysis using fisher vectors. *ACM Transactions on Graphics (Proc SIGGRAPH)*, *33*(6), 174:1–174:9.

Shotton, J., Blake, A., & Cipolla, R. (2008). Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(7), 1270–1281.

Thayananthan, A., Stenger, B., Torr, PHS., Cipolla, R. (2003). shape context and chamfer matching in cluttered scenes. In: CVPR, pp. 127–133

Wang, S., Zhang, L., Liang, Y., Pan, Q. (2012). Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In: CVPR, pp. 2216–2223

Wang, X., & Tang, X. (2009). Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(11), 1955–1967.

Winnemöller, H. (2011). XDoG: Advanced image stylization with extended difference-of-gaussians. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, pp. 147–156

Wu, Y. N., Si, Z., Gong, H., & Zhu, S. C. (2010). Learning active basis model for object detection and recognition. *International Journal of Computer Vision*, *90*(2), 198–235.

Xu, Z., Chen, H., Zhu, S. C., & Luo, J. (2008). A hierarchical compositional model for face representation and sketching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(6), 955–969.

Yu, Q., Yang, Y., Song, Y., Xiang, T., Hospedales, T. (2015). Sketch-a-net that beats humans. In: Proceedings of the British Machine Vision Conference (BMVC)

Zelnik-Manor, L., Perona, P. (2004). Self-tuning spectral clustering. In: NIPS, pp. 1601–1608